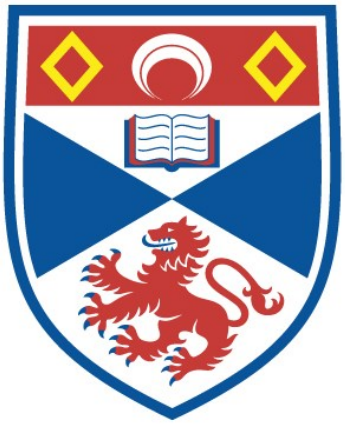
SOME APPLICATIONS OF COMPUTER ALGEBRA AND INTERVAL MATHEMATICS

Mansor Bin Monsi

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



1988

Some Applications of Computer Algebra and Interval Mathematics

MANSOR BIN MONSI

Thesis submitted for the degree of Doctor of Philosophy

of the University of St Andrews

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

I hereby certify that the candidate has fulfilled the conditions of Resolution and Regulations appropriate to the degree of Ph.D. of the University of St Andrews and that he is qualified to submit this thesis in application for that degree.

I Mansor Bin Monsi hereby certify that this thesis has been written by me, that it is a record of work carried out by me from February 1984 to December 1987, and that it has not been submitted in any previous application for a higher degree.

Acknowledgements

I would like to thank my supervisor Mr. M. A. Wolfe for all his help and encouragement during my time as a research student. I am grateful to the Universiti Pertanian Malaysia and to the Jabatan Perkhidmatan Awam Malaysia for their financial support.

ABSTRACT

This thesis contains some applications of Computer Algebra to unconstrained optimization and some applications of Interval Mathematics to the problem of simultaneously bounding the simple zeros of polynomials.

Chapter 1 contains a brief introduction to Computer Algebra and Interval Mathematics, and several of the fundamental results from Interval Mathematics which are used in Chapters 4 and 5.

Chapter 2 contains a survey of those features of the symbol manipulation package *ALGLIB*[SheW--85] which it is necessary to understand in order to use *ALGLIB* as explained in Chapter 3.

Chapter 3 contains a description of Sisser's method [Sis---82a] for unconstrained minimization and several modifications thereof which are implemented using the pseudo-code of Dennis and Schnabel [DenS--83], and *ALGLIB*. Chapter 3 also contains numerical results corresponding to Sisser's method and its modifications for 7 examples.

Chapter 4 contains a new algorithm PRSS for the simultaneous estimation of polynomial zeros and the corresponding interval form IRSS for simultaneously bounding real polynomial zeros. Comparisons are made with some related existing algorithms. Numerical results of the comparisons are also given in this chapter.

Chapter 5 contains an application of an idea due to Neumaier [Neu---85] to the problem of constructing interval versions of point iterative procedures for the estimation of simple zeros of analytic functions. In particular, interval versions of some point iterative procedures for the simultaneous estimation of simple (complex) polynomial zeros are described. Finally, numerical results are given to show the efficiency of the new algorithm.

CONTENTS

CHAPTER 1

Computer Algebra and Interval Mathematics

1.1 Introduction

In many algorithms for the numerical solution of problems in Applied Mathematics it is necessary to perform tedious algebra or to determine formulae for the first and second partial derivatives of functions which are represented by complicated expressions in several variables. Nowadays [SheW--87] it is possible to use computer software in the form of symbol-manipulation packages (software systems which perform algebraic manipulation, analytical differentiation, *etc*) for these purposes. One such symbol-manipulation package is *ALGLIB* [SheW--85], the use of which is described in Chapter 2.

Chapter 3 contains a description of some modifications of Sisser's method [Sis---82a] combined with the use of procedures available in [DenS--83] for unconstrained optimization in which *ALGLIB* is used to compute the gradient and the Hessian of the objective function.

Algorithms for the numerical solution of problems in Applied Mathematics are usually intended to produce a set of real numbers which in some sense represent an estimate of the exact solution. The difference between the exact solution and the numerical solution is due to rounding error and truncation error and possibily to data error. Rounding error results from the fact that real numbers cannot, in general, be represented exactly in a computer, and truncation error results from analytical approximations which are necessary in algorithms for the numerical solution of nonlinear problems.

In practice it is difficult or impossible to obtain realistic bounds on the error in the numerical solution of a given problem if real machine arithmetic is used [Van---78]. It is, however,

possible to obtain rigorous bounds on the exact solution of a given problem if machine interval arithmetic [Moo---79] [AleH--83] is used.

Most algorithms for the numerical solution of real nonlinear problems are iterative and are used to generate sequences of real numbers, vectors, or matrices which converge to a unique solution from a given initial estimate under appropriate hypotheses. It is often difficult and computationally expensive to verify that the hypotheses corresponding to the existence, uniqueness, and convergence theorems for such numerical algorithms hold; it is often impossible to do so with complete rigour because of rounding error. Nevertheless it is possible, at least in principle, to write a program for the solution of a given numerical problem which, given the problem functions, the data, and the data-tolerances, (*a*) checks for the ***non-existence*** of a solution; (*b*) checks for the ***existence*** of a solution; (*c*) checks for the ***uniqueness*** of a solution; (*d*) determines bounds on the solution which are ***computationally rigorous*** and are often ***arbitrarily sharp*** to within the precision of the machine arithmetic and data tolerances by using interval mathematics, a subject which began seriously to be studied after the appearance of the Ph.D thesis of R.E. Moore in 1962 [Moo---62].

Chapters 4 and 5 contain a description of some algorithms for estimating and bounding the zeros of polynomials simultaneously. These algorithms are implemented in Triplex S-algol [ColM--82b], [McbWs-83], a high level programming language which supports machine interval arithmetic.

1.2 Computer Algebra

Computer algebra is that part of computer science in which the design, analysis, computer implementation, and application of algorithms for performing symbolic mathematical operations such as occur in algebra and analysis are studied [SheW--87].

A symbol manipulation package is a software system which implements one or more computer algebra algorithms.

The importance of computer algebra to applied mathematicians lies in the symbol manipulation packages which are currently available. Symbol manipulation packages make possible the transfer of large amounts of mathematical ability to the user; they are also able to perform very rapidly large amounts of tedious algebra without the errors which are usually produced by human beings.

Several symbol manipulation packages are currently available, MACSYMA [Bog---77] being probably the most versatile. The symbol manipulation package *ALGLIB* [SheW--85], although having only a few of the capabilities of MACSYMA, is easily interfaced with programs in S-algol and Triplex S-algol, and is therefore used extensively in the work which is described in this thesis.

1.3 Interval Mathematics

The extension of the ***real interval*** (§1.3.1) is the ***rectangular complex interval*** which is stated in §1.3.2. Both topics are widely discussed in [AleH--83] (see also [Moo---79]). The information given in this section is used in Chapters 4 and 5.

1.3.1 Real Intervals

The fundamental entity in interval mathematics is the bounded closed ***real interval*** $\underline{x} = [x_I, x_S]$ where $x_I \in R$ is the ***infimum*** of $\underline{x}$ and $x_S \in R$ is the ***supremum*** of $\underline{x}$. The set $I(R)$ of real intervals is defined by

$$I(R) = \{\underline{x} = [x_I, x_S] \mid x_I, x_S \in R \wedge x_I \leq x_S\}. \qquad 1.3.1.1$$

Definition 1.3.1.1

Let $\underline{x}, \underline{y} \in I(R)$ be given. Then $\underline{x} = \underline{y}$ if and only if $x_I = y_I$ and $x_S = y_S$. □

Definition 1.3.1.2

Let $\underline{x}, \underline{y} \in I(R)$ be given. Then

$$(\underline{x} \subset \underline{y}) \Leftrightarrow (y_I \leq x_I \leq x_S < y_S \vee y_I < x_I \leq x_S \leq y_S \vee y_I < x_I \leq x_S < y_S),$$

$$(\underline{x} \subseteq \underline{y}) \Leftrightarrow (y_I \leq x_I \leq x_S \leq y_S),$$

$$(x \in \underline{x}) \Leftrightarrow (x_I \leq x \leq x_S),$$

$$(\underline{x} < \underline{y}) \Leftrightarrow (x_S < y_I),$$

and if $a, b \in R$ then

$$(\underline{x} < a) \Leftrightarrow (x_S < a),$$

and

$$(b < \underline{y}) \Leftrightarrow (b < y_I).$$ □

Definition 1.3.1.3

Let $* \in \{+, -, \cdot, /\}$ be a binary operation on R, and let $\underline{x}, \underline{y} \in I(R)$ be given. Then

$$\underline{x} * \underline{y} = \{x * y \mid x \in \underline{x} \wedge y \in \underline{y}\},$$

save that if $0 \in \underline{y}$ then $\underline{x}/\underline{y}$ is not defined. □

From Definition 1.3.1.3 it follows that

$$\underline{x} + \underline{y} = [x_I + y_I, x_S + y_S], \qquad 1.3.1.2$$

$$\underline{x} - \underline{y} = [x_I - y_S, x_S - y_I], \qquad 1.3.1.3$$

$$\underline{x} \cdot \underline{y} = [\min\{x_I y_I, x_I y_S, x_S y_I, x_S y_S\},$$
$$\max\{x_I y_I, x_I y_S, x_S y_I, x_S y_S\}], \qquad 1.3.1.4$$

and if $0 \notin \underline{y}$ then

$$\underline{x}/\underline{y} = [\min\{x_I/y_S, x_I/y_I, x_S/y_S, x_S/y_I\},$$
$$\max\{x_I/y_S, x_I/y_I, x_S/y_S, x_S/y_I\}]. \qquad 1.3.1.5$$

Definition 1.3.1.4

An interval $\underline{x} \in I(R)$ is *degenerate* (or is a *point interval*) if and only if

$$x_I = x_S. \quad \square$$

The set $I_D(R)$ of degenerate intervals and the set R of real numbers are isomorphic. This permits a meaning to be given to $x * \underline{y}$ $(x \in R, \underline{y} \in I(R), * \in \{+, -, \cdot, /\})$.

Definition 1.3.1.5

If $x \in R$ and $\underline{y} \in I(R)$ then

$$x + \underline{y} = [x + y_I, x + y_S],$$
$$x - \underline{y} = [x - y_S, x - y_I],$$
$$x\underline{y} = [\min\{xy_I, xy_S\}, \max\{xy_I, xy_S\}],$$

and if $0 \notin \underline{y}$ then

$$x/\underline{y} = [\min\{x/y_S, x/y_I\}, \max\{x/y_S, x/y_I\}]. \quad \square$$

Definition 1.3.1.5 is consistent with 1.3.1.2–1.3.1.5.

Definition 1.3.1.6

The interval $\underline{x} \in I(R)$ is *symmetric* if and only if $x_I = -x_S$. $\square$

By Definition 1.3.1.6, all symmetric intervals $\underline{x} \in I(R)$ are of the form $[-x, x]$ for some $x \in R$ such that $0 \leq x$.

Definition 1.3.1.7

The *magnitude* $|\underline{x}|$ of $\underline{x} \in I(R)$ is defined by

$$|\underline{x}| = \max\{|x| \mid x \in \underline{x}\}. \square$$

Definition 1.3.1.8

If $\underline{x} \in I(R)$ then $-\underline{x} \in I(R)$ is defined by

$$-\underline{x} = [-x_S, -x_I]. \square$$

By Definition 1.3.1.6, $\underline{x} \in I(R)$ is symmetric if and only if $\underline{x} = -\underline{x}$.

Proposition 1.3.1.1

Interval arithmetic is *inclusion monotonic*; that is to say, if $\underline{a}, \underline{b}, \underline{c}, \underline{d} \in I(R)$ then $(\forall * \in \{+, -, \cdot, /\})$

$$(\underline{a} \subseteq \underline{c} \wedge \underline{b} \subseteq \underline{d}) \Rightarrow (\underline{a} * \underline{b} \subseteq \underline{c} * \underline{d}).$$

Proof

By Definition 1.3.1.3

$$\begin{aligned} \underline{a} * \underline{b} &= \{a * b \mid a \in \underline{a} \wedge b \in \underline{b}\} \\ &\subseteq \{c * d \mid c \in \underline{c} \wedge d \in \underline{d}\} \\ &= \underline{c} * \underline{d}. \square \end{aligned}$$

Definition 1.3.1.9

Let $\underline{x}, \underline{y} \in I(R)$ be given. Then the *intersection* $\underline{x} \cap \underline{y}$ of $\underline{x}$ and $\underline{y}$ is defined by

$$\underline{x} \cap \underline{y} = \{z \in R \mid z \in \underline{x} \wedge z \in \underline{y}\}. \square$$

Proposition 1.3.1.2

(*a*) $(\forall \underline{x}, \underline{y} \in I(R))\ \underline{x} \cap \underline{y} = \underline{y} \cap \underline{x}$;

(*b*) $(\forall \underline{x}, \underline{y} \in I(R))\ \underline{x} \cap \underline{y} \subseteq \underline{x} \wedge \underline{x} \cap \underline{y} \subseteq \underline{y}$;

(*c*) $\left(\underline{x} \cap \underline{y} = \underline{x} \Leftrightarrow \underline{x} \subseteq \underline{y}\right) \wedge \left(\underline{x} \cap \underline{y} = \underline{y} \Leftrightarrow \underline{y} \subseteq \underline{x}\right)$.

Proof

(*a*) By Definition 1.3.1.9

$$\begin{aligned} \underline{x} \cap \underline{y} &= \{z \in R \mid z \in \underline{x} \wedge z \in \underline{y}\} \\ &= \{z \in R \mid z \in \underline{y} \wedge z \in \underline{x}\} \\ &= \underline{y} \cap \underline{x}. \end{aligned}$$

(*b*) By Definition 1.3.1.9

$$\underline{x} \cap \underline{y} = \{z \in R \mid z \in \underline{x} \wedge z \in \underline{y}\}.$$

So

$$(z \in \underline{x} \cap \underline{y}) \Rightarrow (z \in \underline{x}),$$

whence

$$\underline{x} \cap \underline{y} \subseteq \underline{x},$$

and

$$(z \in \underline{x} \cap \underline{y}) \Rightarrow (z \in \underline{y}),$$

whence

$$\underline{x} \cap \underline{y} \subseteq \underline{y}.$$

(*c*) By (*b*) $\underline{x} \cap \underline{y} \subseteq \underline{y}$, so

$$(\underline{x} \cap \underline{y} = \underline{x}) \Rightarrow (\underline{x} \subseteq \underline{y}).$$

Conversely, if $\underline{x} \subseteq \underline{y}$ then by Definition 1.3.1.9

$$\underline{x} \cap \underline{y} = \{z \mid z \in \underline{x} \wedge z \in \underline{y}\}$$
$$= \{z \mid z \in \underline{x}\}$$
$$= \underline{x}.$$

Therefore

$$(\underline{x} \cap \underline{y} = \underline{x}) \Leftrightarrow (\underline{x} \subseteq \underline{y}).$$

Interchanging $\underline{x}$ and $\underline{y}$ and using (a), it follows that

$$(\underline{x} \cap \underline{y} = \underline{y}) \Leftrightarrow (\underline{y} \subseteq \underline{x}). \square$$

Proposition 1.3.1.3

If $\underline{x}, \underline{y} \in I(R)$ are given then

$$\underline{x} \cap \underline{y} = \begin{cases} \emptyset & (x_S < y_I \vee y_S < x_I), \\ [\max\{x_I, y_I\}, \min\{x_S, y_S\}] & \text{(otherwise).} \end{cases}$$

Proof

Now

$$\underline{x} \cap \underline{y} = \{z \in R \mid z \in \underline{x} \wedge z \in \underline{y}\}$$
$$= \{z \in R \mid x_I \leq z \leq x_S \wedge y_I \leq z \leq y_S\}.$$

So

$$(x_S < y_I \vee y_S < x_I) \Rightarrow (\nexists z, x_I \leq z \leq x_S \wedge y_I \leq z \leq y_S)$$
$$\Rightarrow (\underline{x} \cap \underline{y} = \emptyset).$$

If $\underline{x} \cap \underline{y} \neq \emptyset$ then there are 6 cases, namely (a) $x_I \leq y_I \leq x_S \leq y_S$, (b) $y_I \leq x_I \leq x_S \leq y_S$, (c) $y_I \leq x_I \leq y_S \leq x_S$ and 3 other cases obtained from (a)–(c) by interchanging $\underline{x}$ and $\underline{y}$.

(a) $\qquad (z \in [\max\{x_I, y_I\}, \min\{x_S, y_S\}]) \Leftrightarrow (z \in [y_I, x_S])$

$$\Leftrightarrow (z \in \underline{y} \wedge z \in \underline{x})$$

$$\Leftrightarrow (z \in \underline{x} \cap \underline{y}).$$

So in this case

$$\underline{x} \cap \underline{y} = [\max\{x_I, y_I\}, \min\{x_S, y_S\}]. \qquad 1.3.1.6$$

(*b*) $(z \in [\max\{x_I, y_I\}, \min\{x_S, y_S\}]) \Leftrightarrow (z \in [x_I, x_S])$

$$\Leftrightarrow (z \in \underline{x} = \underline{x} \cap \underline{y}).$$

So 1.3.1.6 holds.

(*c*) $(z \in [\max\{x_I, y_I\}, \min\{x_S, y_S\}]) \Leftrightarrow (z \in [x_I, y_S])$

$$\Leftrightarrow (z \in \underline{x} \wedge z \in \underline{y})$$

$$\Leftrightarrow (z \in \underline{x} \cap \underline{y}).$$

Again 1.3.1.6 holds. The 3 other cases are established by interchanging $\underline{x}$ and $\underline{y}$ in the cases (a)–(c). □

Proposition 1.3.1.4

If $\underline{x}, \underline{y}, \underline{z} \in I(R)$ then

(*a*) $\underline{x} + (\underline{y} + \underline{z}) = (\underline{x} + \underline{y}) + \underline{z}$ (associativity of addition);

(*b*) $\underline{x}(\underline{y}\,\underline{z}) = (\underline{x}\,\underline{y})\underline{z}$ (associativity of multiplication);

(*c*) $\underline{x} + \underline{y} = \underline{y} + \underline{x}$ (commutativity of addition);

(*d*) $\underline{x}\,\underline{y} = \underline{y}\,\underline{x}$ (commutativity of multiplication).

Proof

The proofs of (a)–(d) follow from Definition 1.3.1.3. □

Proposition 1.3.1.5

If $\underline{0} = [0, 0]$ and $\underline{1} = [1, 1]$. Then

(*a*) $(\underline{x} + \underline{y} = \underline{y} \quad (\forall \underline{y} \in I(R)) \Leftrightarrow (\underline{x} = \underline{0})$;

(*b*) $(\underline{x}\,\underline{y} = \underline{y} \quad (\forall \underline{y} \in I(R)) \Leftrightarrow (\underline{x} = \underline{1})$.

Proof

(*a*) $$(\underline{x} = \underline{0} \wedge \underline{y} \in I(R)) \Rightarrow (\underline{x} + \underline{y} = [0 + y_I, 0 + y_S] = \underline{y}).$$

Conversely, suppose that

$$\underline{x} + \underline{y} = \underline{y}\,(\forall \underline{y} \in I(R)).$$

Then, setting $\underline{y} = \underline{0}$,

$$\underline{x} + \underline{0} = \underline{0}$$

whence $\underline{x} = \underline{0}$. So

$$(\underline{x} + \underline{y} = \underline{y}\,(\forall \underline{y} \in I(R))) \Rightarrow (\underline{x} = \underline{0}).$$

(*b*) Suppose that $\underline{x} = \underline{1}$. Then $(\forall \underline{y} \in I(R))$

$$\begin{aligned} \underline{x}\,\underline{y} &= \{xy \mid x \in \underline{1} \wedge y \in \underline{y}\} \\ &= \{y \mid y \in \underline{y}\} \\ &= \underline{y}. \end{aligned}$$

So

$$(\underline{x} = \underline{1}) \Rightarrow (\underline{x}\,\underline{y} = \underline{y}).$$

Conversely, suppose that

$$\underline{x}\,\underline{y} = \underline{y}\,(\forall \underline{y} \in I(R)).$$

Then, in particular $\underline{x}\,\underline{y} = \underline{y}$ holds with $\underline{y} = \underline{1}$, whence $\underline{x} = \underline{1}$. So

$$(\underline{x}\,\underline{y} = \underline{y}\,(\forall \underline{y} \in I(R))) \Rightarrow (\underline{x} = \underline{1}).$$ □

Proposition 1.3.1.6

$$((\underline{x}, \underline{y} \in I(R)) \wedge (\underline{x}\,\underline{y} = \underline{0})) \Rightarrow ((\underline{x} = \underline{0}) \vee (\underline{y} = \underline{0})).$$

Proof

$$\begin{aligned}(\underline{x}\,\underline{y} = \underline{0}) &\Rightarrow (\{xy \mid x \in \underline{x} \wedge y \in \underline{y}\} = \{0\}) \\ &\Rightarrow (xy = 0\ (\forall x \in \underline{x})(\forall y \in \underline{y})) \\ &\Rightarrow (x = 0\ (\forall x \in \underline{x}) \vee y = 0\ (\forall y \in \underline{y})) \\ &\Rightarrow (\underline{x} = \underline{0} \vee \underline{y} = \underline{0}).\end{aligned}$$ □

Proposition 1.3.1.7

Interval arithmetic is subdistributive; that is to say $(\forall \underline{x}, \underline{y}, \underline{z} \in I(R))$

$$\underline{x}(\underline{y} + \underline{z}) \subseteq \underline{x}\,\underline{y} + \underline{x}\,\underline{z}.$$

Proof

$$\begin{aligned}\underline{x}(\underline{y} + \underline{z}) &= \{x(y+z) \mid x \in \underline{x} \wedge y \in \underline{y} \wedge z \in \underline{z}\} \\ &= \{xy + xz \mid x \in \underline{x} \wedge y \in \underline{y} \wedge z \in \underline{z}\} \\ &\subseteq \{x'y + x''z \mid x', x'' \in \underline{x}, y \in \underline{y}, z \in \underline{z}\} \\ &= \underline{x}\,\underline{y} + \underline{x}\,\underline{z}.\end{aligned}$$ □

Proposition 1.3.1.8

Let $\underline{x}, \underline{y}, \underline{z} \in I(R)$ be given. Then

(*a*) $(\underline{x} \pm \underline{z} = \underline{y} \pm \underline{z}) \Rightarrow (\underline{x} = \underline{y})$;

(*b*) $(\underline{x}\,\underline{z} = \underline{y}\,\underline{z}) \not\Rightarrow (\underline{x} = \underline{y})$;

(*c*) $(\underline{x}/\underline{z} = \underline{y}/\underline{z}) \Rightarrow (\underline{x} = \underline{y})$.

Proof

(*a*) $$(\underline{x}+\underline{z}=\underline{y}+\underline{z}) \Rightarrow (x_I+z_I=y_I+z_I \wedge x_S+z_S=y_S+z_S)$$
$$\Rightarrow (x_I=y_I \wedge x_S=y_S)$$
$$\Rightarrow (\underline{x}=\underline{y}).$$

Also,

$$(\underline{x}-\underline{z}=\underline{y}-\underline{z}) \Rightarrow (x_I-z_S=y_I-z_S \wedge x_S-z_I=y_S-z_I)$$
$$\Rightarrow (x_I=y_I \wedge x_S=y_S)$$
$$\Rightarrow (\underline{x}=\underline{y}).$$

(*b*) Let $\underline{x}=[1,2]$, $\underline{y}=[-2,0]$, $\underline{z}=[-1,1]$. Then $\underline{x}\,\underline{z}=[-2,2]$ and $\underline{y}\,\underline{z}=[-2,2]$, so $\underline{x}\,\underline{z}=\underline{y}\,\underline{z}$, but $\underline{x}\neq\underline{y}$.

(*c*) If $\underline{x}/\underline{z}$ is defined then $0\notin\underline{z}$. So either $0<z_I$ or $z_S<0$. If $0<z_I$ then

$$\underline{x}/\underline{z}=[\min\{x_I/z_I,x_I/z_S\},\max\{x_S/z_I,x_S/z_S\}]$$

$$=\begin{cases}[x_I/z_S,x_S/z_I] & 0\le x_I,\\ [x_I/z_I,x_S/z_S] & x_S\le 0,\\ [x_I/z_I,x_S/z_I] & x_I\le 0\le x_S.\end{cases}$$

So

$$(0<y_I \wedge \underline{x}/\underline{z}=\underline{y}/\underline{z}) \Rightarrow \left(\frac{x_I}{z_S}=\frac{y_I}{z_S} \wedge \frac{x_S}{z_I}=\frac{y_S}{z_I}\right)$$
$$\Rightarrow (\underline{x}=\underline{y}),$$

$$(y_S<0 \wedge \underline{x}/\underline{z}=\underline{y}/\underline{z}) \Rightarrow \left(\frac{x_I}{z_I}=\frac{y_I}{z_I} \wedge \frac{x_S}{z_S}=\frac{y_S}{z_S}\right)$$
$$\Rightarrow (\underline{x}=\underline{y}),$$

and

$$(y_I < 0 < y_S \wedge \underline{x}/\underline{z} = \underline{y}/\underline{z}) \Rightarrow (\frac{x_I}{z_I} = \frac{y_I}{z_I} \wedge \frac{x_S}{z_I} = \frac{y_S}{z_I})$$
$$\Rightarrow (\underline{x} = \underline{y}).$$

The case $z_S < 0$ is similar. So $(\underline{x}/\underline{z} = \underline{y}/\underline{z}) \Rightarrow (\underline{x} = \underline{y})$. □

Definition 1.3.1.10

The *width* $w(\underline{x})$ of $\underline{x} \in I(R)$ is defined by $w(\underline{x}) = x_S - x_I$. □

Definition 1.3.1.11

The *midpoint* $m(\underline{x})$ of $\underline{x} \in I(R)$ is defined by $m(\underline{x}) = \frac{1}{2}(x_I + x_S)$. □

Proposition 1.3.1.9

Let $\underline{x}, \underline{y} \in I(R)$ be given. Then $(\underline{x} \subseteq \underline{y}) \Rightarrow (|\underline{x}| \leq |\underline{y}|)$, but the converse is not in general true.

Proof

If $\underline{x} \subseteq \underline{y}$ then $y_I \leq x_I \leq x_S \leq y_S$. So

(*a*) If $0 \leq y_I$ then $|\underline{y}| = y_S \geq x_S = |\underline{x}|$.

(*b*) If $y_I \leq 0 \leq x_I$ then $|\underline{y}| = \max\{|y_I|, y_S\} \geq x_S = |\underline{x}|$.

(*c*) If $x_I \leq 0 \leq x_S$ then $|\underline{y}| = \max\{|y_I|, y_S\} \geq \max\{|x_I|, x_S\} = |\underline{x}|$.

(*d*) If $x_S \leq 0 \leq y_S$ then $|\underline{y}| = \max\{|y_I|, y_S\} \geq |x_I| = |\underline{x}|$.

(*e*) If $y_S \leq 0$ then $|\underline{y}| = |y_I| \geq |x_I| = |\underline{x}|$.

Therefore $(\underline{x} \subseteq \underline{y}) \Rightarrow (|\underline{x}| \leq |\underline{y}|)$. Conversely, $(\underline{x} = [1, 2]) \Rightarrow (|\underline{x}| = 2)$, and $(\underline{y} = [3, 4]) \Rightarrow (|\underline{y}| = 4)$. So $|\underline{x}| < |\underline{y}|$ but clearly $\underline{x} \not\subseteq \underline{y}$. □

Proposition 1.3.1.10

(*a*) $|\underline{x}| \geq 0$ $(\forall \underline{x} \in I(R))$ and $(|\underline{x}| = \underline{0}) \Leftrightarrow (\underline{x} = \underline{0})$;

(*b*) $|\underline{x} \pm \underline{y}| \leq |\underline{x}| + |\underline{y}|$ $(\forall \underline{x}, \underline{y} \in I(R))$;

(*c*) $|\alpha \underline{x}| = |\alpha||\underline{x}|$ $(\forall \alpha \in R)(\forall \underline{x} \in I(R))$;

(*d*) $|\underline{x}\,\underline{y}| = |\underline{x}||\underline{y}|$ $(\forall \underline{x}, \underline{y} \in I(R))$.

Proof

(*a*) Clearly $|\underline{x}| = \max\{|x_I|, |x_S|\} \geq 0$. Also

$$\begin{aligned}(|\underline{x}| = 0) &\Leftrightarrow (\max\{|x_I|, |x_S|\} = 0)\\ &\Leftrightarrow (|x_I| = 0 \wedge |x_S| = 0)\\ &\Leftrightarrow (x_I = 0 \wedge x_S = 0)\\ &\Leftrightarrow (\underline{x} = \underline{0}).\end{aligned}$$

(*b*) By 1.3.1.2 and 1.3.1.3, and Definition 1.3.1.7

$$\begin{aligned}|\underline{x} + \underline{y}| &= \max\{|x_I + y_I|, |x_S + y_S|\}\\ &\leq \max\{|x_I| + |y_I|, |x_S| + |y_S|\}\\ &\leq \max\{|x_I|, |x_S|\} + \max\{|y_I|, |y_S|\}\\ &= |\underline{x}| + |\underline{y}|.\end{aligned}$$

Also,

$$\begin{aligned}|\underline{x} - \underline{y}| &= \max\{|x_I - y_S|, |x_S - y_I|\}\\ &\leq \max\{|x_I| + |y_S|, |x_S| + |y_I|\}\\ &\leq \max\{|x_I|, |x_S|\} + \max\{|y_I|, |y_S|\}\\ &= |\underline{x}| + |\underline{y}|.\end{aligned}$$

(*c*) By Definitions 1.3.1.5 and 1.3.1.7

$$\begin{aligned}|\alpha\underline{x}| &= \max\{|\alpha x_I|, |\alpha x_S|\} \\ &= \max\{|\alpha||x_I|, |\alpha||x_S|\} \\ &= |\alpha| \max\{|x_I|, |x_S|\} \\ &= |\alpha||\underline{x}|.\end{aligned}$$

(*d*) By Definition 1.3.1.7

$$\begin{aligned}|\underline{x}\,\underline{y}| &= \max\{|xy| \mid x \in \underline{x} \wedge y \in \underline{y}\} \\ &= \max\{|x||y| \mid x \in \underline{x} \wedge y \in \underline{y}\} \\ &= \max\{|x| \mid x \in \underline{x}\} \max\{|y| \mid y \in \underline{y}\} \\ &= |\underline{x}||\underline{y}|. \quad \square\end{aligned}$$

Proposition 1.3.1.11

$(\forall \underline{x} \in I(R))$

$$w(\underline{x}) = \max\{|\hat{x} - \tilde{x}| \mid \hat{x}, \tilde{x} \in \underline{x}\}.$$

Proof

Let $\hat{x}, \tilde{x} \in \underline{x}$ be given, and suppose, without loss of generality, that $\hat{x} \geq \tilde{x}$. Then $x_I \leq \tilde{x} \leq \hat{x} \leq x_S$, whence

$$\begin{aligned}|\hat{x} - \tilde{x}| &= \hat{x} - \tilde{x} \\ &\leq x_S - x_I \\ &= w(\underline{x}),\end{aligned}$$

with equality if and only if $\hat{x} = x_S$ and $\tilde{x} = x_I$. Therefore

$$w(\underline{x}) = \max\{|\hat{x} - \tilde{x}| \mid \hat{x}, \tilde{x} \in \underline{x}\}. \square$$

Proposition 1.3.1.12

Let $\underline{x}, \underline{y} \in I(R)$ be given. Then $(\underline{x} \subseteq \underline{y}) \Rightarrow (w(\underline{x}) \leq w(\underline{y}))$. But the converse is not in general true.

Proof

$$\begin{aligned}(\underline{x} \subseteq \underline{y}) &\Rightarrow (y_I \leq x_I \leq x_S \leq y_S) \\ &\Rightarrow (x_S - x_I \leq y_S - y_I) \\ &\Rightarrow (w(\underline{x}) \leq w(\underline{y})).\end{aligned}$$

Conversely, $(\underline{x} = [1, 2]) \Rightarrow (w(\underline{x}) = 1)$, and $(\underline{y} = [3, 5]) \Rightarrow (w(\underline{y}) = 2)$. Clearly $w(\underline{x}) \leq w(\underline{y})$ but $\underline{x} \not\subseteq \underline{y}$. $\square$

Proposition 1.3.1.13

(*a*) $w(\underline{x}) \geq 0$ and $(w(\underline{x}) = 0) \Leftrightarrow (\underline{x}$ is degenerate);

(*b*) $w(\underline{x} \pm \underline{y}) = w(\underline{x}) + w(\underline{y})$ $(\forall \underline{x}, \underline{y} \in I(R))$;

(*c*) $w(\alpha \underline{x}) = |\alpha| w(\underline{x})$ $(\forall \alpha \in R)$ $(\forall \underline{x} \in I(R)$.

Proof

(*a*) Clearly $w(\underline{x}) = x_S - x_I \geq 0$. Also,

$$\begin{aligned}(w(\underline{x}) = 0) &\Leftrightarrow (x_I = x_S) \\ &\Leftrightarrow (\underline{x} \text{ is degenerate}).\end{aligned}$$

(*b*) By 1.3.1.2,1.3.1.3 and Definition 1.3.1.10

$$w(\underline{x} + \underline{y}) = (x_S + y_S) - (x_I + y_I)$$

$$= (x_S - x_I) + (y_S - y_I)$$

$$= w(\underline{x}) + w(\underline{y}),$$

and

$$w(\underline{x} - \underline{y}) = (x_S - y_I) - (x_I - y_S)$$

$$= (x_S - x_I) + (y_S - y_I)$$

$$= w(\underline{x}) + w(\underline{y}).$$

(c) By Proposition 1.3.1.11

$$w(\alpha\underline{x}) = \max\{|\alpha\hat{x} - \alpha\tilde{x}| \mid \hat{x}, \tilde{x} \in \underline{x}\}$$

$$= \max\{|\alpha||\hat{x} - \tilde{x}| \mid \hat{x}, \tilde{x} \in \underline{x}\}$$

$$= |\alpha| \max\{|\hat{x} - \tilde{x}| \mid \hat{x}, \tilde{x} \in \underline{x}\}$$

$$= |\alpha| w(\underline{x}). \quad \square$$

Proposition 1.3.1.14

$(\forall \alpha, \beta \in R)(\forall \underline{x}, \underline{y} \in I(R))$

$$w(\alpha\underline{x} \pm \beta\underline{y}) = |\alpha| w(\underline{x}) + |\beta| w(\underline{y}).$$

Proof

By Proposition 1.3.1.13(b), (c)

$$w(\alpha\underline{x} \pm \beta\underline{y}) = w(\alpha\underline{x}) + w(\beta\underline{y})$$

$$= |\alpha| w(\underline{x}) + |\beta| w(\underline{y}). \quad \square$$

Proposition 1.3.1.15

$(\forall \underline{x}, \underline{y} \in I(R))$

(a) $w(\underline{x}\,\underline{y}) \leq w(\underline{x})|\underline{y}| + |\underline{x}| w(\underline{y})$;

(b) $w(\underline{x}\,\underline{y}) \geq \max\{w(\underline{x})|\underline{y}|, |\underline{x}| w(\underline{y})\}$.

Proof

(*a*) By Proposition 1.3.1.11 and Definition1.3.1.3

$$\begin{aligned} w(\underline{x}\,\underline{y}) &= \max\{|x'y' - x''y''| \mid x', x'' \in \underline{x} \wedge y', y'' \in \underline{y}\} \\ &= \max\{|x'(y' - y'') + y''(x' - x'')| \mid x', x'' \in \underline{x} \wedge y', y'' \in \underline{y}\} \\ &\leq \max\{|x'||y' - y''| + |y''||x' - x''| \mid x', x'' \in \underline{x} \wedge y', y'' \in \underline{y}\} \\ &= \max\{|x'| \mid x' \in \underline{x}\} \max\{|y' - y''| \mid y', y'' \in \underline{y}\} + \\ &\quad \max\{|y''| \mid y'' \in \underline{y}\} \max\{|x' - x''| \mid x', x'' \in \underline{x}\} \\ &= |\underline{x}|w(\underline{y}) + |\underline{y}|w(\underline{x}). \end{aligned}$$

(*b*) By Proposition 1.3.1.11 and Definition 1.3.1.3

$$\begin{aligned} w(\underline{x}\,\underline{y}) &= \max\{|x'y' - x''y''| \mid x', x'' \in \underline{x} \wedge y', y'' \in \underline{y}\} \\ &\geq \max\{|xy' - xy''| \mid x \in \underline{x} \wedge y', y'' \in \underline{y}\} \\ &= \max\{|x||y' - y''| \mid x \in \underline{x} \wedge y', y'' \in \underline{y}\} \\ &= \max\{|x| \mid x \in \underline{x}\} \max\{|y' - y''| \mid y', y'' \in \underline{y}\} \\ &= |\underline{x}|w(\underline{y}). \end{aligned}$$

Interchanging $\underline{x}$ and $\underline{y}$ we also have $w(\underline{x}\,\underline{y}) \geq |\underline{y}|w(\underline{x})$. So

$$w(\underline{x}\,\underline{y}) \geq \max\{|\underline{x}|w(\underline{y}), w(\underline{x})|\underline{y}|\}. \quad \square$$

Definition 1.3.1.12

Let $f : D \subseteq R^1 \to R^1$ be a given function. The function $\underline{f} : I(D) \to I(R)$ is an ***interval extension*** of $f : D \to R$ if and only if $\underline{f}(x) = \underline{f}([x, x]) = f(x)$ $(\forall x \in D)$. The function $f : D \to R$ is called the ***real restriction*** of $\underline{f} : I(D) \to I(R)$. $\square$

Definition 1.3.1.13

Let $\underline{f} : I(D) \to I(R)$ be a given function. Then $\underline{f}$ is *inclusion monotonic* if and only if $\underline{f}(\underline{x}) \subseteq \underline{f}(\underline{y})$ $(\forall \underline{x}, \underline{y} \in I(D)$ such that $\underline{x} \subseteq \underline{y})$. □

Definition 1.3.1.14

Let $f : D \subseteq R \to R$ be a given function which is continuous in $\hat{D} \subseteq D$. Then the function $\bar{\underline{f}} : I(\hat{D}) \to I(R)$ defined by

$$\bar{\underline{f}}(\underline{x}) = \left\{ f(x) \mid x \in \underline{x} \wedge \underline{x} \in I(\hat{D}) \right\} \qquad 1.3.1.7$$

is called the *united extension* of $f : \hat{D} \to R$. □

Proposition 1.3.1.16

Functions $\underline{f} : I(R) \to I(R)$ are not, in general,

(a) inclusion monotonic;

(b) united extensions of functions $f : R \to R$.

Proof

(a) Let $\underline{f} : I(R) \to I(R)$ be defined by

$$\underline{f}(\underline{x}) = m(\underline{x}) + \tfrac{1}{2}(\underline{x} - m(\underline{x})).$$

Then

$$\begin{aligned} \underline{f}([0,2]) &= 1 + \tfrac{1}{2}[-1,1] \\ &= [\tfrac{1}{2}, \tfrac{3}{2}], \end{aligned}$$

and

$$\begin{aligned} \underline{f}([0,1]) &= \tfrac{1}{2} + \tfrac{1}{2}[-\tfrac{1}{2}, \tfrac{1}{2}] \\ &= [\tfrac{1}{4}, \tfrac{3}{4}]. \end{aligned}$$

So

$$\underline{f}([0,1]) \not\subseteq \underline{f}([0,2])$$

even though $[0,1] \subseteq [0,2]$.

(*b*) Let $\underline{f} : I(R) \to I(R)$ be defined by

$$\underline{f}(\underline{x}) = \underline{x} - \underline{x}.$$

Then $\underline{f}([0,0]) = 0$, so if $f : R \to R$ is the real restriction of $\underline{f}$ then $f(x) = 0$ $(\forall x \in \underline{x})$ $(\forall \underline{x} \in I(R))$. So $f(x) = 0$ $(\forall x \in R)$. So $(\forall \underline{x} \in I(R))$,

$$\{f(x) \mid x \in \underline{x}\} = \{0\}.$$

But

$$\begin{aligned} \underline{f}(\underline{x}) &= \underline{x} - \underline{x} \\ &= [x_I - x_S, x_S - x_I] \\ &\neq [0,0] \end{aligned}$$

for all non-degenerate intervals $\underline{x}$. So

$$\underline{f}(\underline{x}) \neq \{f(x) \mid x \in \underline{x}\}.$$

Therefore $\underline{f}$ is not the united extension of f. □

Proposition 1.3.1.17

Let $f : D \subseteq R \to R$ be a given function which is continuous in $\hat{D} \subseteq D$, and let $\bar{\underline{f}} : I(\hat{D}) \to I(R)$ be the united extension of $f : \hat{D} \to R$. Then $\bar{\underline{f}}$ is inclusion monotonic.

Proof

Suppose that $(\forall \underline{x}, \underline{y} \in I(\hat{D}))$, $\underline{x} \subseteq \underline{y}$. Then by 1.3.1.7

$$\begin{aligned} \underline{\bar{f}}(\underline{x}) &= \{ f(x) \mid x \in \underline{x} \} \\ &\subseteq \{ f(x) \mid x \in \underline{y} \} \\ &= \underline{\bar{f}}(\underline{y}). \end{aligned}$$ □

Proposition 1.3.1.18

Let $f : D \subseteq R \to R$ be a continuous function. If $\underline{f} : I(D) \to I(R)$ is an inclusion monotonic interval extension of f, then

(*a*) $f(x) \in \underline{f}(\underline{x})$ $(\forall x \in \underline{x} \wedge \underline{x} \in I(D))$;

(*b*) $\underline{\bar{f}}(\underline{x}) \subseteq \underline{f}(\underline{x})$ $(\forall \underline{x} \in I(R))$.

Proof

(*a*) Suppose that $x \in \underline{x}$. Then $[x, x] \subseteq \underline{x}$. So by inclusion monotonicity of $\underline{f}$, $\underline{f}([x, x])$ $\subseteq \underline{f}(\underline{x})$. But since $\underline{f}$ is an interval extension of f, it follows that $\underline{f}([x, x]) = f(x)$. Therefore $f(x) \in \underline{f}(\underline{x})$.

(*b*) Suppose that $y \in \underline{\bar{f}}(\underline{x})$. Then $\exists x \in \underline{x}$ such that $y = f(x)$. By (*a*) $f(x) \in \underline{f}(\underline{x})$. So $y \in \underline{f}(\underline{x})$. Therefore $\underline{\bar{f}}(\underline{x}) \subseteq \underline{f}(\underline{x})$. □

Proposition 1.3.1.19

If $\underline{f} : I(R) \to I(R)$ and $\underline{g} : I(R) \to I(R)$ are inclusion monotonic and $\underline{h} : I(R) \to I(R)$ is defined by

$$\underline{h}(\underline{x}) = \underline{f}(\underline{x}) * \underline{g}(\underline{x}),$$

where $* \in \{+, -, \cdot, /\}$, then $\underline{h}$ is inclusion monotonic.

Proof

By Proposition 1.3.1.1

$$(\underline{x} \subseteq \underline{y}) \Rightarrow (\underline{f}(\underline{x}) \subseteq \underline{f}(\underline{y}) \wedge \underline{g}(\underline{x}) \subseteq \underline{g}(\underline{y}))$$
$$\Rightarrow (\underline{f}(\underline{x}) * \underline{g}(\underline{x}) \subseteq \underline{f}(\underline{y}) * \underline{g}(\underline{y}))$$
$$\Rightarrow (\underline{h}(\underline{x}) \subseteq \underline{h}(\underline{y})). \square$$

1.3.2 Rectangular Complex Intervals

Definition 1.3.2.1

Let $\underline{x}_R, \underline{x}_I \in I(R)$ be given. Then the set $\underline{x}$ defined by

$$\underline{x} = \{x_R + ix_I \in C \mid x_R \in \underline{x}_R \wedge x_I \in \underline{x}_I\},$$

where $i = \sqrt{-1}$ is called a ***rectangular complex interval***, or a ***rectangle*** when no ambiguity is possible. The symbol $I_R(C)$ denotes the set of rectangular complex intervals. □

Definition 1.3.2.2

Let $\underline{x} = \underline{x}_R + i\underline{x}_I \in I_R(C)$ and $\underline{y} = \underline{y}_R + i\underline{y}_I \in I_R(C)$ be given. Then $\underline{x} = \underline{y}$ if and only if $\underline{x}_R = \underline{y}_R$ and $\underline{x}_I = \underline{y}_I$. □

Definition 1.3.2.3

Let $\underline{x} = \underline{x}_R + i\underline{x}_I \in I_R(C)$ and $\underline{y} = \underline{y}_R + i\underline{y}_I \in I_R(C)$ be given. Then $\underline{x} \subseteq \underline{y}$ if and only if $\underline{x}_R \subseteq \underline{y}_R$ and $\underline{x}_I \subseteq \underline{y}_I$. □

Definition 1.3.2.4

If $\underline{x} = \underline{x}_R + i\underline{x}_I \in I_R(C)$ and $\underline{y} = \underline{y}_R + i\underline{y}_I \in I_R(C)$ then

$$\underline{x}+\underline{y}=(\underline{x}_R+\underline{y}_R)+i(\underline{x}_I+\underline{y}_I),$$

$$\underline{x}-\underline{y}=(\underline{x}_R-\underline{y}_R)+i(\underline{x}_I-\underline{y}_I),$$

$$\underline{x}\,\underline{y}=(\underline{x}_R\underline{y}_R-\underline{x}_I\underline{y}_I)+i(\underline{x}_R\underline{y}_I+\underline{x}_I\underline{y}_R),$$

and if $0 \notin \underline{y}_R^2+\underline{y}_I^2$, where $(\forall \underline{a} \in I(R))$

$$\underline{a}^2=\{a^2 \mid a \in \underline{a}\}$$

$$=\begin{cases}[a_I^2, a_S^2] & (0 \le a_I)\\ [0, \max\{a_I^2, a_S^2\}] & (a_I < 0 < a_S)\\ [a_S^2, a_I^2] & (a_S \le 0)\end{cases},$$

then

$$\underline{x}/\underline{y}=(\underline{x}_R\underline{y}_R+\underline{x}_I\underline{y}_I)/(\underline{y}_R^2+\underline{y}_I^2)$$
$$+i\{(\underline{x}_I\underline{y}_R-\underline{x}_R\underline{y}_I)/(\underline{y}_R^2+\underline{y}_I^2)\}. \square$$

Definition 1.3.2.5

The interval $\underline{x}=\underline{x}_R+i\underline{x}_I \in I_R(C)$ is ***degenerate*** (or is a ***point interval***) if and only if $\underline{x}_R$ and $\underline{x}_I$ are ***degenerate***. $\square$

By Definition 1.3.2.5 degenerate rectangular complex intervals are of the form $[x_R, x_R]+i[x_I, x_I]$, where $x_R, x_I \in R$. If $\underline{x}=[x_R, x_R]+i[x_I, x_I]$, and $\underline{y}=[y_R, y_R]+i[y_I, y_I]$, then by Definition 1.3.2.4,

$$\underline{x}+\underline{y}=[x_R+y_R, x_R+y_R]+i[x_I+y_I, x_I+y_I],$$

$$\underline{x}-\underline{y}=[x_R-y_R, x_R-y_R]+i[x_I-y_I, x_I-y_I],$$

$$\underline{x}\,\underline{y}=[x_Ry_R-x_Iy_I, x_Ry_R-x_Iy_I]+i[x_Ry_I+x_Iy_R, x_Ry_I+x_Iy_R],$$

$$\underline{x}/\underline{y}=[x_Ry_R+x_Iy_I, x_Ry_R+x_Iy_I]/[y_R^2+y_I^2, y_R^2+y_I^2]$$

$$+ i[x_I y_R - x_R y_I, x_I y_R - x_R y_I]/[y_R^2 + y_I^2, y_R^2 + y_I^2].$$

Now if $z_1 = x_R + ix_I \in C$ and $z_2 = y_R + iy_I \in C$, then

$$\begin{aligned}
z_1 + z_2 &= (x_R + y_R) + i(x_I + y_I), \\
z_1 - z_2 &= (x_R - y_R) + i(x_I - y_I), \\
z_1 z_2 &= (x_R + ix_I)(y_R + iy_I) \\
&= (x_R y_R - x_I y_I) + i(x_I y_R + x_R y_I), \\
z_1/z_2 &= (x_R + ix_I)/(y_R + iy_I) \\
&= (x_R + ix_I)(y_R - iy_I)/(y_R^2 + y_I^2) \\
&= \{(x_R y_R + x_I y_I) + i(x_I y_R - x_R y_I)\}/(y_R^2 + y_I^2).
\end{aligned}$$

So it is clear that the set of degenerate rectangular complex intervals and the set C of complex numbers are isomorphic. Thus the set of degenerate rectangular complex intervals may be formally indentified with C. This permits a meaning to be given to $z * \underline{x}$, where $z \in C$, $\underline{x} \in I_R(C)$, and $* \in \{+, -, \cdot, /\}$ in which $* : I_R(C) \times I_R(C) \to I_R(C)$ is defined by Definition 1.3.2.4; we have

$$z * \underline{x} = ([z_R, z_R] + i[z_I, z_I]) * (\underline{x}_R + i\underline{x}_I).$$

Definition 1.3.2.6

Let $\underline{x} = \underline{x}_R + i\underline{x}_I \in I_R(C)$ and $\underline{y} = \underline{y}_R + i\underline{y}_I \in I_R(C)$ be given. Then the *intersection* $\underline{x} \cap \underline{y}$ of $\underline{x}$ and $\underline{y}$ is defined by

$$\underline{x} \cap \underline{y} = (\underline{x}_R \cap \underline{y}_R) + i(\underline{x}_I \cap \underline{y}_I). \quad \square$$

Proposition 1.3.2.1

(*a*) $(\forall \underline{x}, \underline{y} \in I_R(C))\ \underline{x} \cap \underline{y} = \underline{y} \cap \underline{x}$;

(*b*) $(\forall \underline{x}, \underline{y} \in I_R(C))\ \underline{x} \cap \underline{y} \subseteq \underline{x} \wedge \underline{x} \cap \underline{y} \subseteq \underline{y}$;

(*c*) $(\underline{x} \cap \underline{y} = \underline{x}) \Leftrightarrow (\underline{x} \subseteq \underline{y}) \wedge (\underline{x} \cap \underline{y} = \underline{y}) \Leftrightarrow (\underline{y} \subseteq \underline{x})$.

Proof

(*a*) By Definition 1.3.2.6 and Proposition 1.3.1.2(*a*)

$$\begin{aligned}\underline{x} \cap \underline{y} &= (\underline{x}_R \cap \underline{y}_R) + i(\underline{x}_I \cap \underline{y}_I) \\ &= (\underline{y}_R \cap \underline{x}_R) + i(\underline{y}_I \cap \underline{x}_I) \\ &= \underline{y} \cap \underline{x}.\end{aligned}$$

(*b*) By Definition 1.3.2.6 and Proposition 1.3.1.2(*b*)

$$\begin{aligned}\underline{x} \cap \underline{y} &= (\underline{x}_R \cap \underline{y}_R) + i(\underline{x}_I \cap \underline{y}_I) \\ &\subseteq \underline{x}_R + i\underline{x}_I \\ &= \underline{x},\end{aligned}$$

and

$$\begin{aligned}\underline{x} \cap \underline{y} &= (\underline{x}_R \cap \underline{y}_R) + i(\underline{x}_I \cap \underline{y}_I) \\ &\subseteq \underline{y}_R + i\underline{y}_I \\ &= \underline{y}.\end{aligned}$$

(*c*) By Definitions 1.3.2.6,1.3.2.3 and Proposition 1.3.1.2(*c*)

$$\begin{aligned}(\underline{x} \cap \underline{y} = \underline{x}) &\Leftrightarrow (\underline{x}_R \cap \underline{y}_R = \underline{x}_R \wedge \underline{x}_I \cap \underline{y}_I = \underline{x}_I) \\ &\Leftrightarrow (\underline{x}_R \subseteq \underline{y}_R \wedge \underline{x}_I \subseteq \underline{y}_I) \\ &\Leftrightarrow (\underline{x} \subseteq \underline{y}).\end{aligned}$$

Also,

$$(\underline{x} \cap \underline{y} = \underline{y}) \Leftrightarrow (\underline{x}_R \cap \underline{y}_R = \underline{y}_R \wedge \underline{x}_I \cap \underline{y}_I = \underline{y}_I)$$

$$\Leftrightarrow (\underline{y}_R \subseteq \underline{x}_R \wedge \underline{y}_I \subseteq \underline{x}_I)$$

$$\Leftrightarrow (\underline{y} \subseteq \underline{x}). \square$$

Proposition 1.3.2.2 (Inclusion Monotonicity)

Let $\underline{x}^{(i)}, \underline{y}^{(i)} \in I_R(C)$ $(i = 1, 2)$ be given. Then $(\forall * \in \{+, -, \cdot, /\})$

$$(\underline{x}^{(i)} \subseteq \underline{y}^{(i)} \ (i = 1, 2)) \Rightarrow (\underline{x}^{(1)} * \underline{x}^{(2)} \subseteq \underline{y}^{(1)} * \underline{y}^{(2)}).$$

Proof

By Proposition 1.3.1.1 and Definition 1.3.2.3

(a)
$$\begin{aligned}\underline{x}^{(1)} + \underline{x}^{(2)} &= (\underline{x}_R^{(1)} + i\underline{x}_I^{(1)}) + (\underline{x}_R^{(2)} + i\underline{x}_I^{(2)}) \\ &= (\underline{x}_R^{(1)} + \underline{x}_R^{(2)}) + i(\underline{x}_I^{(1)} + \underline{x}_I^{(2)}) \\ &\subseteq (\underline{y}_R^{(1)} + \underline{y}_R^{(2)}) + i(\underline{y}_I^{(1)} + \underline{y}_I^{(2)}) \\ &= (\underline{y}_R^{(1)} + i\underline{y}_I^{(1)}) + (\underline{y}_R^{(2)} + i\underline{y}_I^{(2)}) \\ &= \underline{y}^{(1)} + \underline{y}^{(2)}.\end{aligned}$$

(b)
$$\begin{aligned}\underline{x}^{(1)} - \underline{x}^{(2)} &= (\underline{x}_R^{(1)} + i\underline{x}_I^{(1)}) - (\underline{x}_R^{(2)} + i\underline{x}_I^{(2)}) \\ &= (\underline{x}_R^{(1)} - \underline{x}_R^{(2)}) + i(\underline{x}_I^{(1)} - \underline{x}_I^{(2)}) \\ &\subseteq (\underline{y}_R^{(1)} - \underline{y}_R^{(2)}) + i(\underline{y}_I^{(1)} - \underline{y}_I^{(2)}) \\ &= (\underline{y}_R^{(1)} + i\underline{y}_I^{(1)}) - (\underline{y}_R^{(2)} + i\underline{y}_I^{(2)}) \\ &= \underline{y}^{(1)} - \underline{y}^{(2)}.\end{aligned}$$

(c)
$$\begin{aligned}\underline{x}^{(1)}\underline{x}^{(2)} &= (\underline{x}_R^{(1)}\underline{x}_R^{(2)} - \underline{x}_I^{(1)}\underline{x}_I^{(2)}) + i(\underline{x}_R^{(1)}\underline{x}_I^{(2)} + \underline{x}_I^{(1)}\underline{x}_R^{(2)}) \\ &\subseteq (\underline{y}_R^{(1)}\underline{y}_R^{(2)} - \underline{y}_I^{(1)}\underline{y}_I^{(2)}) + i(\underline{y}_R^{(1)}\underline{y}_I^{(2)} + \underline{y}_I^{(1)}\underline{y}_R^{(2)}) \\ &= \underline{y}^{(1)}\underline{y}^{(2)}.\end{aligned}$$

(*d*) If $0 \notin \underline{y}^{(2)}$ then $0 \notin \underline{x}^{(2)}$ so both $\underline{x}^{(1)}/\underline{x}^{(2)}$ and $\underline{y}^{(1)}/\underline{y}^{(2)}$ are defined and by Proposition 1.3.1.1

$$\begin{aligned}
\underline{x}^{(1)}/\underline{x}^{(2)} &= (\underline{x}_R^{(1)}\underline{x}_R^{(2)} + \underline{x}_I^{(1)}\underline{x}_I^{(2)})/(\underline{x}_R^{(2)^2} + \underline{x}_I^{(2)^2}) \\
&\quad + i(\underline{x}_I^{(1)}\underline{x}_R^{(2)} - \underline{x}_R^{(1)}\underline{x}_I^{(2)})/(\underline{x}_R^{(2)^2} + \underline{x}_I^{(2)^2}) \\
&\subseteq (\underline{y}_R^{(1)}\underline{y}_R^{(2)} + \underline{y}_I^{(1)}\underline{y}_I^{(2)})/(\underline{y}_R^{(2)^2} + \underline{y}_I^{(2)^2}) \\
&\quad + i(\underline{y}_I^{(1)}\underline{y}_R^{(2)} - \underline{y}_R^{(1)}\underline{y}_I^{(2)})/(\underline{y}_R^{(2)^2} + \underline{y}_I^{(2)^2}) \\
&= \underline{y}^{(1)}/\underline{y}^{(2)}. \quad \square
\end{aligned}$$

Proposition 1.3.2.3

If $\underline{x}, \underline{y}, \underline{z} \in I_R(C)$ then

(*a*) $\underline{x} + (\underline{y} + \underline{z}) = (\underline{x} + \underline{y}) + \underline{z}$ (associativity of addition);

(*b*) $\underline{x}(\underline{y}\,\underline{z}) \neq (\underline{x}\,\underline{y})\underline{z}$ (associativity of multiplication is not in general valid);

(*c*) $\underline{x} + \underline{y} = \underline{y} + \underline{x}$ (commutativity of addition);

(*d*) $\underline{x}\,\underline{y} = \underline{y}\,\underline{x}$ (commutativity of multiplication).

Proof

Suppose that $\underline{x} = \underline{x}_R + i\underline{x}_I$, $\underline{y} = \underline{y}_R + i\underline{y}_I$ and $\underline{z} = \underline{z}_R + i\underline{z}_I$.

(*a*) By Proposition 1.3.1.4(a)

$$\begin{aligned}
\underline{x} + (\underline{y} + \underline{z}) &= \underline{x} + \{(\underline{y}_R + \underline{z}_R) + i(\underline{y}_I + \underline{z}_I)\} \\
&= \underline{x}_R + i\underline{x}_I + \{(\underline{y}_R + \underline{z}_R) + i(\underline{y}_I + \underline{z}_I)\} \\
&= \{\underline{x}_R + (\underline{y}_R + \underline{z}_R)\} + i\{\underline{x}_I + (\underline{y}_I + \underline{z}_I)\} \\
&= \{(\underline{x}_R + \underline{y}_R) + \underline{z}_R\} + i\{(\underline{x}_I + \underline{y}_I) + \underline{z}_I\} \\
&= \{(\underline{x}_R + \underline{y}_R) + i(\underline{x}_I + \underline{y}_I)\} + \underline{z}_R + i\underline{z}_I \\
&= (\underline{x} + \underline{y}) + \underline{z}.
\end{aligned}$$

(*b*) Let $\underline{x} = [2, 4] + i[0, 0]$, $\underline{y} = \underline{z} = [1, 1] + i[1, 1]$. Then

$$\begin{aligned}\underline{x}\,\underline{y} &= ([2, 4][1, 1] - [0, 0][1, 1]) + i([2, 4][1, 1] + [0, 0][1, 1]) \\ &= [2, 4] + i[2, 4].\end{aligned}$$

So

$$\begin{aligned}(\underline{x}\,\underline{y})\underline{z} &= ([2, 4][1, 1] - [2, 4][1, 1]) + i([2, 4][1, 1] + [2, 4][1, 1]) \\ &= ([2, 4] - [2, 4]) + i([2, 4] + [2, 4]) \\ &= [-2, 2] + i[4, 8]\end{aligned}$$

and

$$\begin{aligned}\underline{y}\,\underline{z} &= ([1, 1][1, 1] - [1, 1][1, 1]) + i([1, 1][1, 1] + [1, 1][1, 1]) \\ &= [0, 0] + i[2, 2].\end{aligned}$$

So

$$\begin{aligned}\underline{x}(\underline{y}\,\underline{z}) &= ([2, 4][0, 0] - [0, 0][2, 2]) + i([2, 4][2, 2] + [0, 0][0, 0]) \\ &= [0, 0] + i[4, 8].\end{aligned}$$

Therefore $(\underline{x}\,\underline{y})\underline{z} \neq \underline{x}(\underline{y}\,\underline{z})$. Therefore it is not in general true that $(\underline{x}\,\underline{y})\underline{z} = \underline{x}(\underline{y}\,\underline{z})$.

(*c*) By Proposition 1.3.1.4(c)

$$\begin{aligned}\underline{x} + \underline{y} &= (\underline{x}_R + i\underline{x}_I) + (\underline{y}_R + i\underline{y}_I) \\ &= (\underline{x}_R + \underline{y}_R) + i(\underline{x}_I + \underline{y}_I) \\ &= (\underline{y}_R + \underline{x}_R) + i(\underline{y}_I + \underline{x}_I) \\ &= \underline{y} + \underline{x}.\end{aligned}$$

(*d*) By Proposition 1.3.1.4(c),(d)

$$\begin{aligned}\underline{x}\,\underline{y} &= (\underline{x}_R\underline{y}_R - \underline{x}_I\underline{y}_I) + i(\underline{x}_R\underline{y}_I + \underline{x}_I\underline{y}_R)\\ &= (\underline{y}_R\underline{x}_R - \underline{y}_I\underline{x}_I) + i(\underline{y}_I\underline{x}_R + \underline{y}_R\underline{x}_I)\\ &= (\underline{y}_R\underline{x}_R - \underline{y}_I\underline{x}_I) + i(\underline{y}_R\underline{x}_I + \underline{y}_I\underline{x}_R)\\ &= \underline{y}\,\underline{x}.\ \square\end{aligned}$$

Proposition 1.3.2.4

If $\underline{0} = [0,0] + i[0,0] \in I_R(C)$ and $\underline{1} = [1,1] + i[0,0] \in I_R(C)$. Then

(*a*) $(\underline{x} + \underline{y} = \underline{x}\ (\forall \underline{x} \in I_R(C))) \Leftrightarrow (\underline{y} = \underline{0})$;

(*b*) $(\underline{x}\,\underline{y} = \underline{x}\ (\forall \underline{x} \in I_R(C))) \Leftrightarrow (\underline{y} = \underline{1})$.

Proof

(*a*) By Proposition 1.3.1.5(*a*) $(\forall \underline{x} \in I_R(C))$

$$\begin{aligned}(\underline{y} = \underline{0}) &\Rightarrow (\underline{x} + \underline{y} = (\underline{x}_R + \underline{0}) + i(\underline{x}_I + \underline{0}))\\ &\Rightarrow (\underline{x} + \underline{y} = \underline{x}_R + i\underline{x}_I)\\ &\Rightarrow (\underline{x} + \underline{y} = \underline{x}).\end{aligned}$$

Conversely, suppose that $\underline{x} + \underline{y} = \underline{x}$ $(\forall \underline{x} \in I_R(C))$ and $\underline{y} \neq \underline{0}$. Then by Proposition 1.3.1.5(*a*)

$$\begin{aligned}(\underline{x} + \underline{y} = \underline{x}) &\Rightarrow ((\underline{x}_R + \underline{y}_R) + i(\underline{x}_I + \underline{y}_I) = \underline{x}_R + \underline{x}_I)\\ &\Rightarrow (\underline{x}_R + \underline{y}_R = \underline{x}_R \wedge \underline{x}_I + \underline{y}_I = \underline{x}_I)\\ &\Rightarrow (\underline{y}_R = \underline{0} \wedge \underline{y}_I = \underline{0})\\ &\Rightarrow (\underline{y} = \underline{0}),\end{aligned}$$

contrary to the hypothesis that $\underline{y} \neq \underline{0}$. So

$$(\underline{x} + \underline{y} = \underline{x}) \Rightarrow (\underline{y} = \underline{0}).$$

(*b*) Suppose that $\underline{y} = \underline{1}$. Then $(\forall \underline{x} \in I_R(C))$

$$\begin{aligned}\underline{x}\,\underline{y} &= (\underline{x}_R[1,1] - \underline{x}_I[0,0]) + i(\underline{x}_R[0,0] + \underline{x}_I[1,1]) \\ &= \underline{x}_R + i\underline{x}_I \\ &= \underline{x}.\end{aligned}$$

Conversely, suppose that $(\forall \underline{x} \in I_R(C))$, $\underline{x}\,\underline{y} = \underline{x}$. Then, in particular $\underline{x}\,\underline{y} = \underline{x}$ holds with $\underline{x} = \underline{1}$, whence $\underline{y} = \underline{1}$. □

Proposition 1.3.2.5

If $\underline{x}, \underline{y} \in I_R(C)$ and $\underline{x}\,\underline{y} = \underline{0}$ then $\underline{x} = \underline{0}$ or $\underline{y} = \underline{0}$.

Proof

By Definition 1.3.1.3

$$(\underline{x}\,\underline{y} = \underline{0}) \Rightarrow (xy = 0\,(\forall x \in \underline{x})(\forall y \in \underline{y})).$$

Now

$$(\underline{x} \neq \underline{0}) \Rightarrow (\exists \hat{x} \in \underline{x}, \hat{x} \neq 0).$$

So

$$\begin{aligned}(\underline{x}\,\underline{y} = \underline{0} \wedge \underline{x} \neq \underline{0}) &\Rightarrow (\hat{x}y = 0\,(\forall y \in \underline{y})) \\ &\Rightarrow (y = 0\,(\forall y \in \underline{y})) \\ &\Rightarrow (\underline{y} = \underline{0}).\end{aligned}$$

Similarly $(\underline{y} \neq \underline{0}) \Rightarrow (\exists \hat{y} \in \underline{y}, \hat{y} \neq 0)$. So $(\underline{x}\,\underline{y} = \underline{0}) \Rightarrow (\underline{x} = \underline{0})$. Therefore

$$(\underline{x}\,\underline{y} = \underline{0}) \Rightarrow (\underline{x} = \underline{0} \vee \underline{y} = \underline{0}).$$ □

Proposition 1.3.2.6

Rectangular complex interval arithmetic is subdistributive; that is to say,

$$\underline{x}(\underline{y}+\underline{z}) \subseteq \underline{x}\,\underline{y}+\underline{x}\,\underline{z} \quad (\forall \underline{x},\underline{y},\underline{z} \in I_R(C))$$

Proof

By Definitions 1.3.2.3, 1.3.2.4 and Propositions 1.3.1.4 and 1.3.1.7,

$$\begin{aligned}
\underline{x}(\underline{y}+\underline{z}) &= \underline{x}\{(\underline{y}_R+\underline{z}_R)+i(\underline{y}_I+\underline{z}_I)\} \\
&= \{\underline{x}_R(\underline{y}_R+\underline{z}_R)-\underline{x}_I(\underline{y}_I+\underline{z}_I)\} \\
&\quad +i\{\underline{x}_R(\underline{y}_I+\underline{z}_I)+\underline{x}_I(\underline{y}_R+\underline{z}_R)\} \\
&\subseteq \{(\underline{x}_R\underline{y}_R+\underline{x}_R\underline{z}_R)\}-(\underline{x}_I\underline{y}_I+\underline{x}_I\underline{z}_I)\} \\
&\quad +i\{(\underline{x}_R\underline{y}_I+\underline{x}_R\underline{z}_I)+(\underline{x}_I\underline{y}_R+\underline{x}_I\underline{z}_R)\} \\
&= \{(\underline{x}_R\underline{y}_R-\underline{x}_I\underline{y}_I)+i(\underline{x}_R\underline{y}_I+\underline{x}_I\underline{y}_R)\} \\
&\quad +\{(\underline{x}_R\underline{z}_R-\underline{x}_I\underline{z}_I)+i(\underline{x}_R\underline{z}_I+\underline{x}_I\underline{z}_R)\} \\
&= \underline{x}\,\underline{y}+\underline{x}\,\underline{z}. \quad \square
\end{aligned}$$

Definition 1.3.2.7

Let $\underline{x}=\underline{x}_R+i\underline{x}_I \in I_R(C)$ be given. Then the ***magnitude*** $|\underline{x}|$ of $\underline{x}$ is defined by

$$|\underline{x}| = |\underline{x}_R|+|\underline{x}_I|. \quad \square$$

Proposition 1.3.2.7

Let $\underline{x},\underline{y} \in I_R(C)$ be given. Then

(*a*) $|\underline{x}| \geq 0$;

(*b*) $(|\underline{x}|=0) \Leftrightarrow (\underline{x}=\underline{0})$;

(*c*) $|\underline{x} \pm \underline{y}| \leq |\underline{x}|+|\underline{y}|$;

(*d*) $|\alpha\underline{x}| \leq |\alpha|_R|\underline{x}|$ $(\forall \alpha \in C)$, where $|\alpha|_R = |\alpha_R|+|\alpha_I|$;

(*e*) $|\underline{x}\,\underline{y}| \leq |\underline{x}||\underline{y}|$.

Proof

(*a*) By Definition 1.3.2.7 and Proposition 1.3.1.10(a)

$$|\underline{x}| = |\underline{x}_R| + |\underline{x}_I| \geq 0.$$

(*b*) By Definition 1.3.2.7 and Proposition 1.3.1.10(a)

$$\begin{aligned}(|\underline{x}| = 0) &\Leftrightarrow (|\underline{x}_R| + |\underline{x}_I| = 0)\\ &\Leftrightarrow (|\underline{x}_R| = 0 \wedge |\underline{x}_I| = 0)\\ &\Leftrightarrow (\underline{x}_R = \underline{0} \wedge \underline{x}_I = \underline{0})\\ &\Leftrightarrow (\underline{x} = \underline{0}).\end{aligned}$$

(*c*) By Definition 1.3.2.7 and Proposition 1.3.1.10(b)

$$\begin{aligned}|\underline{x} + \underline{y}| &= |\underline{x}_R + \underline{y}_R| + |\underline{x}_I + \underline{y}_I|\\ &\leq |\underline{x}_R| + |\underline{y}_R| + |\underline{x}_I| + |\underline{y}_I|\\ &= (|\underline{x}_R| + |\underline{x}_I|) + (|\underline{y}_R| + |\underline{y}_I|)\\ &= |\underline{x}| + |\underline{y}|.\end{aligned}$$

Also,

$$\begin{aligned}|\underline{x} - \underline{y}| &= |\underline{x}_R - \underline{y}_R| + |\underline{x}_I - \underline{y}_I|\\ &\leq |\underline{x}_R| + |\underline{y}_R| + |\underline{x}_I| + |\underline{y}_I|\\ &= (|\underline{x}_R| + |\underline{x}_I|) + (|\underline{y}_R| + |\underline{y}_I|)\\ &= |\underline{x}| + |\underline{y}|.\end{aligned}$$

(*d*) By Proposition 1.3.1.10(b),(c), and Definition 1.3.2.7

$$|\alpha\underline{x}| = |(\alpha_R\underline{x}_R - \alpha_I\underline{x}_I) + i(\alpha_R\underline{x}_I + \alpha_I\underline{x}_R)|$$

$$= |\alpha_R \underline{x}_R - \alpha_I \underline{x}_I| + |\alpha_R \underline{x}_I + \alpha_I \underline{x}_R|$$

$$\leq |\alpha_R \underline{x}_R| + |\alpha_I \underline{x}_I| + |\alpha_R \underline{x}_I| + |\alpha_I \underline{x}_R|$$

$$= |\alpha_R||\underline{x}_R| + |\alpha_I||\underline{x}_I| + |\alpha_R||\underline{x}_I| + |\alpha_I||\underline{x}_R|$$

$$= (|\alpha_R| + |\alpha_I|)(|\underline{x}_R| + |\underline{x}_I|)$$

$$= |\alpha|_R|\underline{x}|.$$

(*e*) By Proposition 1.3.1.10(b),(d), and Definition 1.3.2.7

$$|\underline{x}\,\underline{y}| = |(\underline{x}_R \underline{y}_R - \underline{x}_I \underline{y}_I) + i(\underline{x}_R \underline{y}_I + \underline{x}_I \underline{y}_R)|$$

$$= |\underline{x}_R \underline{y}_R - \underline{x}_I \underline{y}_I| + |\underline{x}_R \underline{y}_I + \underline{x}_I \underline{y}_R|$$

$$\leq |\underline{x}_R \underline{y}_R| + |\underline{x}_I \underline{y}_I| + |\underline{x}_R \underline{y}_I| + |\underline{x}_I \underline{y}_R|$$

$$= |\underline{x}_R||\underline{y}_R| + |\underline{x}_I||\underline{y}_I| + |\underline{x}_R||\underline{y}_I| + |\underline{x}_I||\underline{y}_R|$$

$$= (|\underline{x}_R| + |\underline{x}_I|)(|\underline{y}_R| + |\underline{y}_I|)$$

$$= |\underline{x}||\underline{y}|. \square$$

Definition 1.3.2.8

Let $\underline{x} = \underline{x}_R + i\underline{x}_I \in I_R(C)$ be given. Then the ***midpoint*** $m(\underline{x})$ of $\underline{x}$ is defined by

$$m(\underline{x}) = m(\underline{x}_R) + im(\underline{x}_I). \square$$

Definition 1.3.2.9

Let $\underline{x} = \underline{x}_R + i\underline{x}_I \in I_R(C)$ be given. Then the ***width*** $w(\underline{x})$ of $\underline{x}$ is defined by

$$w(\underline{x}) = w(\underline{x}_R) + w(\underline{x}_I). \square$$

Proposition 1.3.2.8

Let $\underline{x}, \underline{y} \in I_R(C)$ be given. Then $(\underline{x} \subseteq \underline{y}) \Rightarrow (w(\underline{x}) \leq w(\underline{y}))$. But the converse is not in general true.

Proof

By Definition 1.3.2.9 and Proposition 1.3.1.12

$$\begin{aligned}
(\underline{x} \subseteq \underline{y}) &\Rightarrow (\underline{x}_R \subseteq \underline{y}_R \wedge \underline{x}_I \subseteq \underline{y}_I) \\
&\Rightarrow (y_{RI} \leq x_{RI} \leq x_{RS} \leq y_{RS} \wedge y_{II} \leq x_{II} \leq x_{IS} \leq y_{IS}) \\
&\Rightarrow (x_{RS} - x_{RI} \leq y_{RS} - y_{RI} \wedge x_{IS} - x_{II} \leq y_{IS} - y_{II}) \\
&\Rightarrow (w(\underline{x}_R) \leq w(\underline{y}_R) \wedge w(\underline{x}_I) \leq w(\underline{y}_I)) \\
&\Rightarrow (w(\underline{x}_R) + w(\underline{x}_I) \leq w(\underline{y}_R) + w(\underline{y}_I)) \\
&\Rightarrow (w(\underline{x}) \leq w(\underline{y})).
\end{aligned}$$

Conversely, we take $\underline{x} = [1,2]+i[1,2]$ and $\underline{y} = [3,5]+i[3,5]$. So $w(\underline{x}) = (2-1)+(2-1) = 2$ and $w(\underline{y}) = (5-3)+(5-3) = 4$. Clearly $w(\underline{x}) < w(\underline{y})$ but $\underline{x} \not\subseteq \underline{y}$. □

Proposition 1.3.2.9

If $\underline{x}, \underline{y} \in I_R(C)$ then

(*a*) $w(\underline{x} \pm \underline{y}) = w(\underline{x}) + w(\underline{y})$;

(*b*) $w(\alpha\underline{x}) = |\alpha|_R w(\underline{x})$ $(\forall \alpha \in C)$;

(*c*) $w(\underline{x}\,\underline{y}) \leq |\underline{x}|w(\underline{y}) + w(\underline{x})|\underline{y}|$;

(*d*) $w(\underline{x}\,\underline{y}) \geq |\underline{x}|w(\underline{y}) \wedge w(\underline{x}\,\underline{y}) \geq |\underline{y}|w(\underline{x})$;

(*e*) $w(\underline{x} \cap \underline{y}) \leq \min\{w(\underline{x}), w(\underline{y})\}$.

Proof

(*a*) By Definition 1.3.2.9 and Proposition 1.3.1.13(b)

$$\begin{aligned}
w(\underline{x} \pm \underline{y}) &= w((\underline{x}_R \pm \underline{y}_R) + i(\underline{x}_I \pm \underline{y}_I)) \\
&= w(\underline{x}_R \pm \underline{y}_R) + w(\underline{x}_I \pm \underline{y}_I) \\
&= w(\underline{x}_R) + w(\underline{y}_R) + w(\underline{x}_I) + w(\underline{y}_I)
\end{aligned}$$

$$= (w(\underline{x}_R) + w(\underline{x}_I)) + (w(\underline{y}_R) + w(\underline{y}_I))$$

$$= w(\underline{x}) + w(\underline{y}).$$

(*b*) By Definition 1.3.2.9 and Proposition 1.3.1.13(*c*)

$$\begin{aligned} w(\alpha\underline{x}) &= w((\alpha_R\underline{x}_R - \alpha_I\underline{x}_I) + i(\alpha_R\underline{x}_I + \alpha_I\underline{x}_R)) \\ &= w(\alpha_R\underline{x}_R - \alpha_I\underline{x}_I) + w(\alpha_R\underline{x}_I + \alpha_I\underline{x}_R) \\ &= w(\alpha_R\underline{x}_R) + w(\alpha_I\underline{x}_I) + w(\alpha_R\underline{x}_I) + w(\alpha_I\underline{x}_R) \\ &= |\alpha_R|w(\underline{x}_R) + |\alpha_I|w(\underline{x}_I) + |\alpha_R|w(\underline{x}_I) + |\alpha_I|w(\underline{x}_R) \\ &= (|\alpha_R| + |\alpha_I|)(w(\underline{x}_R) + w(\underline{x}_I)) \\ &= |\alpha|_R w(\underline{x}). \end{aligned}$$

(*c*) By Definition 1.3.2.9 and Propositions 1.3.1.15(*a*), 1.3.1.13(*b*)

$$\begin{aligned} w(\underline{x}\,\underline{y}) &= w(\underline{x}_R\underline{y}_R - \underline{x}_I\underline{y}_I) + w(\underline{x}_R\underline{y}_I + \underline{x}_I\underline{y}_R) \\ &= w(\underline{x}_R\underline{y}_R) + w(\underline{x}_I\underline{y}_I) + w(\underline{x}_R\underline{y}_I) + w(\underline{x}_I\underline{y}_R) \\ &\le |\underline{x}_R|w(\underline{y}_R) + w(\underline{x}_R)|\underline{y}_R| + |\underline{x}_I|w(\underline{y}_I) + w(\underline{x}_I)|\underline{y}_I| \\ &\quad + |\underline{x}_R|w(\underline{y}_I) + w(\underline{x}_R)|\underline{y}_I| + |\underline{x}_I|w(\underline{y}_R) + w(\underline{x}_I)|\underline{y}_R| \\ &= (|\underline{x}_R| + |\underline{x}_I|)(w(\underline{y}_R) + w(\underline{y}_I)) \\ &\quad + (|\underline{y}_R| + |\underline{y}_I|)(w(\underline{x}_R) + w(\underline{x}_I)) \\ &= |\underline{x}|w(\underline{y}) + |\underline{y}|w(\underline{x}). \end{aligned}$$

(*d*) By Proposition 1.3.1.15(*b*)

$$\begin{aligned} w(\underline{x}\,\underline{y}) &= w(\underline{x}_R\underline{y}_R) + w(\underline{x}_I\underline{y}_I) + w(\underline{x}_R\underline{y}_I) + w(\underline{x}_I\underline{y}_R) \\ &\ge |\underline{x}_R|w(\underline{y}_R) + |\underline{x}_I|w(\underline{y}_I) + |\underline{x}_R|w(\underline{y}_I) + |\underline{x}_I|w(\underline{y}_R) \end{aligned}$$

$$= (|\underline{x}_R| + |\underline{x}_I|)(w(\underline{y}_R) + w(\underline{y}_I))$$
$$= |\underline{x}|w(\underline{y}).$$

Similarly, by interchanging $\underline{x}$ and $\underline{y}$ we have

$$w(\underline{x}\,\underline{y}) \geq |\underline{y}|w(\underline{x}).$$

(*e*) By Definition 1.3.2.3 and Proposition 1.3.1.12,

$$(\underline{x} \subseteq \underline{y}) \Rightarrow (\underline{x}_R \subseteq \underline{y}_R \wedge \underline{x}_I \subseteq \underline{y}_I).$$

So

$$w(\underline{x}_R \cap \underline{y}_R) + w(\underline{x}_I \cap \underline{y}_I) \leq w(\underline{y}_R) + w(\underline{y}_I) = w(\underline{y}).$$

Similarly,

$$(\underline{y} \subseteq \underline{x}) \Rightarrow (\underline{y}_R \subseteq \underline{x}_R \wedge \underline{y}_I \subseteq \underline{x}_I).$$

So

$$w(\underline{x}_R \cap \underline{y}_R) + w(\underline{x}_I + \underline{y}_I) \leq w(\underline{x}_R) + w(\underline{x}_I) = w(\underline{x}).$$

Therefore by Definitions 1.3.2.6 and 1.3.2.9, $(\forall \underline{x}, \underline{y} \in I_R(C))$

$$w(\underline{x} \cap \underline{y}) = w((\underline{x}_R \cap \underline{y}_R) + i(\underline{x}_I \cap \underline{y}_I))$$
$$= w(\underline{x}_R \cap \underline{y}_R) + w(\underline{x}_I \cap \underline{y}_I)$$
$$\leq \min\{w(\underline{x}), w(\underline{y})\}. \square$$

CHAPTER 2

On Using the Package $\mathcal{ALGLIB}$

2.1 Computable Factorable Functions

The $\mathcal{ALGLIB}$ package manipulates the set of computable factorable functions — a subset of the set of factorable functions described by Mc Cormick [McC---83].

Definition 2.1.1

Let X be the set R or the set $I(R)$. The function $f : X^n \rightarrow X$ is a ***computable factorable function*** (CFF) if and only if the expression $f(x)$, where $x = (x_1, \ldots, x_n)^T \in X^n$ can be represented as the last in a finite sequence of expressions $f_j(x)$ in which

$$f_j(x) = x_j \quad (j = 1, \ldots, n), \tag{2.1.1}$$

and if $j > n$ then

$$f_j(x) = f_k(x) * f_l(x) \quad (k, l < j) \quad (* \in \{+, -, \cdot, /\}), \tag{2.1.2}$$

or

$$f_j(x) = T(f_k(x)) \quad (k < j), \tag{2.1.3}$$

where $T(\cdot) \in \mathcal{F} = \{-(\cdot), sqrt(\cdot), exp(\cdot), ln(\cdot), sin(\cdot), cos(\cdot), atan(\cdot), |\cdot|, (\cdot)^m (m \in Z)\}$ if $X = R$, and $T(\cdot)$ is a continuous inclusion monotonic interval extension of one of the given set of real functions if $X = I(R)$. □

Let $f : X^n \to X^1$ and $g : X^n \to X^1$, where $X = R$ or $X = I(R)$ be given CFFs. $\mathcal{ALGLIB}$ can *combine* f and g to give the CFF $h : X^n \to X^1$ defined by

$$h(x) = f(x) * g(x),$$

where $* \in \{+, -, \cdot, /\}$. Furthermore $\mathcal{ALGLIB}$ can combine $c \in X$ with $f : X^n \to X^1$ to give the CFFs $p : X^n \to X^1$ and $q : X^n \to X^1$ defined by

$$p(x) = c * f(x)$$

and

$$q(x) = f(x) * c,$$

where $* \in \{+, -, \cdot, /\}$.

Let $f : X^n \to X^1$ and $g_i : X^n \to X^1$ $(i = 1, \ldots, n)$ be given CFFs and let $g : X^n \to X^n$ be defined by

$$g(x) = (g_1(x), \ldots, g_n(x))^T.$$

The $\mathcal{ALGLIB}$ can *compose* f with g to give the CFF $h : X^n \to X^1$ defined by

$$h(x) = f(g(x))$$

$$= f(g_1(x), \ldots, g_n(x)).$$

Let $f : X^n \to X^1$ be a given CFF. Then $\mathcal{ALGLIB}$ can determine mixed partial derivatives

of f of any order, and can also determine the gradient $\nabla f : X^n \rightarrow X^n$ and the Hessian $\nabla^2 f : X^n \rightarrow X^{n \times n}$ of f defined by

$$\nabla f(x) = \left(\partial_i f(x)\right)_{n \times 1}$$

and

$$\nabla^2 f(x) = \left(\partial_i \partial_j f(x)\right)_{n \times n},$$

where $\partial_i f = \partial f / \partial x_i$ and $\partial_i \partial_j f = \partial^2 f / \partial x_i \partial x_j \quad (i, j = 1, \ldots, n)$.

2.2 Algebra and Calculus

The use of *ALGLIB* to perform algebraic manipulation and differentiation is described in this section. It is assumed that an S-algol implementation of *ALGLIB* is available and that the program in which *ALGLIB* is to be used is also written in S-algol.

Initially, the names of the variables which are to be used in the expressions which *ALGLIB* is to manipulate must be known. Suppose that the variables x_1 and x_2 are to be used. A vector of strings containing the variable names is created by the statement

```
let names = @1 of string["x1", "x2"]        2.2.1
```

in which *names*(1) points to the string "$x1$" and *names*(2) points to the string "$x2$". The data structures corresponding to the variables "x_1" and "x_2" are created by using the *ALGLIB* procedure *define.variables* through the statement

```
let variables = define.variables(names)     2.2.2
```

The entity ***variables*** is a vector of pointers to the data-structures corresponding to the variables x_1 and x_2.

If the CFFs $f : R^2 \rightarrow R^1$ and $g : R^2 \rightarrow R^1$ are defined by

$$f(x) = x_1 exp(x_2) \qquad 2.2.3$$

and

$$g(x) = x_1^2 + x_2^2, \qquad 2.2.4$$

then the strings ***f.string*** and ***g.string*** representing the expressions $f(x)$ and $g(x)$ respectively may be created by using the statements

$$\underline{let}\ f.string = \text{“}x1 * exp(x2)\text{”} \qquad 2.2.5$$

and

$$\underline{let}\ g.string = \text{“}x1\hat{}2 + x2\hat{}2\text{”} \qquad 2.2.6$$

and the corresponding $\mathcal{ALGLIB}$ data-structures are created by invoking the $\mathcal{ALGLIB}$ procedure ***string.to.function*** as follows.

$$\underline{let}\ f := string.to.function(variables, f.string) \qquad 2.2.7$$

$$\underline{let}\ g := string.to.function(variables, g.string) \qquad 2.2.8$$

The entities f and g are pointers to the data-structures corresponding to the CFFs f and g respectively. The assignment operator := declares f and g to be of type variable poin-

ter, so that f and g may be assigned to subsequently. The assignment operator = declares $f.string$ and $g.string$ to be of type constant string so that neither $f.string$ nor $g.string$ may be assigned to subsequently.

All of the data-structures which are created by $\mathcal{ALGLIB}$ are added to linked lists of the appropriate kind. If it is desired to add the data-structures pointed to by f and g to the appropriate linked lists after first simplifying them, then the $\mathcal{ALGLIB}$ procedure *add.to.list* may be invoked as follows.

$$f := add.to.list(f) \tag{2.2.9}$$

$$g := add.to.list(g) \tag{2.2.10}$$

If the CFFs $h : R^2 \rightarrow R^1$, $p : R^2 \rightarrow R^1$, and $q : R^2 \rightarrow R^1$ are defined by

$$\begin{aligned} h(x) &= f(x)g(x) \\ &= (x_1^2 + x_2^2)x_1 exp(x_2), \end{aligned} \tag{2.2.11}$$

$$\begin{aligned} p(x) &= 1 - f(x), \\ &= 1 - x_1 exp(x_2), \end{aligned} \tag{2.2.12}$$

and

$$\begin{aligned} q(x) &= g(x) - 1 \\ &= x_1^2 + x_2^2 - 1, \end{aligned} \tag{2.2.13}$$

then the corresponding $\mathcal{ALGLIB}$ data-structures and pointers h, p, and q to them are obtained by invoking the $\mathcal{ALGLIB}$ procedures *function.op.function*, *real.op.function*,

and ***function.op.real*** as follows.

$$\underline{let}\ h = function.op.function(f, \text{“}*\text{”}, g) \qquad 2.2.14$$

$$\underline{let}\ p = real.op.function(1, \text{“}-\text{”}, f) \qquad 2.2.15$$

$$\underline{let}\ q = function.op.real(g, \text{“}-\text{”}, 1) \qquad 2.2.16$$

The procedures ***function.op.function***, ***real.op.function***, and ***function.op.real*** all simplify the data-structures which they create before adding them to the appropriate linked list.

Let $u : R^2 \rightarrow R^1$ be defined by

$$\begin{aligned} u(x) &= sin(f(x)) \\ &= sin(x_1 exp(x_2)). \end{aligned} \qquad 2.2.17$$

Then the $\mathcal{ALGLIB}$ data-structure corresponding to the expression $u(x)$ and a pointer u to the data-structure are created by invoking the $\mathcal{ALGLIB}$ procedure ***op.function*** as follows.

$$\underline{let}\ u = op.function(\text{“}sin\text{”}, f) \qquad 2.2.18$$

The procedure ***op.function*** simplifies the data-structure which it creates before adding it to the appropriate linked list. Let $r : R^2 \rightarrow R^2$ be defined by

$$\begin{aligned} r(x) &= (p(x), q(x))^T \\ &= (1 - f(x), g(x) - 1)^T \end{aligned}$$

$$= (1 - x_1 exp(x_2), x_1^2 + x_2^2 - 1)^T, \qquad 2.2.19$$

and let $v : R^2 \rightarrow R^1$ be defined by

$$\begin{aligned} v(x) &= f(r(x)) \\ &= p(x)exp(q(x)) \\ &= \{1 - x_1 exp(x_2)\}exp(x_1^2 + x_2^2 - 1). \end{aligned} \qquad 2.2.20$$

Then a pointer r to the vector of pointers $(p, q)^T$ is created by using the S-algol statement

$$\underline{let}\ r = @1\ \underline{of}\ \underline{pntr}[p, q] \qquad 2.2.21$$

and the $\mathcal{ALGLIB}$ data-structure corresponding to the expression $v(x)$ and a pointer v to the data-structure are created by invoking the $\mathcal{ALGLIB}$ procedure ***compose*** as follows.

$$\underline{let}\ v = compose(f, r) \qquad 2.2.22$$

The procedure ***compose*** simplifies the data-structure which it creates before adding it to the appropriate linked list.

The $\mathcal{ALGLIB}$ data-structures corresponding to the partial derivatives of CFFs may be created by invoking the $\mathcal{ALGLIB}$ procedure ***partial***. For example, the data-structures corresponding to $\partial_1 f(x)$ and to $\partial_2 \partial_1 f(x)$ and the pointers $d1f$ and $d2\,d1f$ to them are created by

$$\underline{let}\ d1f = partial(f, variables(1)) \qquad 2.2.23$$

$$\underline{let}\ d2\,d1f = partial(d1f, variables(2)) \qquad 2.2.24$$

while the data-structures corresponding to the gradient and to the Hessian of f and pointers $grad.f$ and $hess.f$ to them are created by invoking the $\mathcal{ALGLIB}$ procedures ***gradient*** and ***hessian*** as follows.

$$\underline{let}\ grad.f = gradient(f) \qquad 2.2.25$$

$$\underline{let}\ hess.f = hessian(f) \qquad 2.2.26$$

Here, $grad.f$ points to a vector of two pointers $grad.f(1)$ and $grad.f(2)$ which point to the data-structures corresponding to $\partial_1 f(x)$ and to $\partial_2 f(x)$ respectively, and $hess.f$ points to a matrix of 4 pointers $hess.f(i,j)$ $(i,j = 1,2)$, where $hess.f(i,j)$ points to the data-structure corresponding to $\partial_j \partial_i f(x)$.

The $\mathcal{ALGLIB}$ data-structure corresponding to the Jacobian $J : R^2 \rightarrow R^{2\times 2}$ of $r : R^2 \rightarrow R^2$

$$J(x) = \left(\partial_j r_i(x)\right)_{2\times 2} \qquad 2.2.27$$

is created by invoking the $\mathcal{ALGLIB}$ procedure $jacobian$ as follows.

$$\underline{let}\ jacob.r = jacobian(r) \qquad 2.2.28$$

Here, $jacob.r$ points to the matrix of 4 pointers $jacob.r(i,j)$ $(i,j = 1,2)$, where $jacob.r(i,j)$ points to the data-structure corresponding to $\partial_j r_i(x)$.

The procedures ***partial***, ***gradient***, ***hessian***, and $jacobian$ all simplify the data-structures which they create before adding them to the appropriate linked list.

The expression corresponding to a given $\mathcal{ALGLIB}$ data-structure may be written by invoking the $\mathcal{ALGLIB}$ procedure ***function.format***, which creates a string corresponding to the given data-structure. The string can then be written using the S-algol *write* clause. For example, the expression corresponding to the CFF f could be written as follows.

$$\underline{write}\ function.format(f)$$

The output is

$$x1 * exp(x2)$$

The vector of expressions corresponding to the data-structures pointed to by a given vector of pointers may be written by invoking the $\mathcal{ALGLIB}$ procedure ***write.expression.vector***. Similarly the matrix of expressions corresponding to the data-structures pointed to by a given matrix of pointers may be written by invoking the $\mathcal{ALGLIB}$ procedure ***write.expression.matrix***.

Expressions corresponding to CFFs may be evaluated by invoking the $\mathcal{ALGLIB}$ procedure ***evaluate***. The following code causes the CFF $f : R^2 \rightarrow R^1$ to be evaluated in real floating point arithmetic with $x_1 = 2$ and $x_2 = 1$.

$$\underline{let}\ x = @1\ \underline{of}\ \underline{real}[2, 1]$$

$$\underline{let}\ f.value := evaluate(f, x)$$

Here ***f.value*** is a variable of type real which has the value $f(2, 1) = 2exp(1)$.

2.3 Hessians as Sums of Dyads

If $f : R^n \rightarrow R^1$ is a given twice differentiable CFF then $\nabla^2 f(x)$ can be expressed as a

sum of dyads (outer products of vectors). Similarly if $F : R^n \rightarrow R^n$ is a given differentiable function and $J : R^n \rightarrow R^{n\times n}$ is defined by

$$J(x) = \left(\partial_j F_i(x)\right)_{n\times n}, \tag{2.3.1}$$

then $J(x)^T J(x)$ can also be expressed as a sum of dyads.

Theorem 2.3.1

If $f : R^n \rightarrow R^1$ is a twice differentiable CFF then $\exists m \in N$, $\gamma_i : R^n \rightarrow R^1$, $a_i : R^n \rightarrow R^n$, $b_i : R^n \rightarrow R^n$, $(i = 1, \ldots, m)$ such that

$$\nabla^2 f(x) = \sum_{i=1}^{m} \gamma_i(x)\left\{a_i(x)b_i(x)^T + b_i(x)a_i(x)^T\right\}. \tag{2.3.2}$$

Proof

Let $\left(f_p(x)\right)$ be a sequence of expressions which are generated according to the rules 2.1.1–2.1.3, and which is such that $f(x)$ is the last element of the sequence $\left(f_p(x)\right)$. It will be shown that $(\forall p \geq 1)$ $\exists m_p \in N$, $\gamma_{pi} : R^n \rightarrow R^1$, $a_{pi} : R^n \rightarrow R^n$, $b_{pi} : R^n \rightarrow R^n$ $(i = 1, \ldots, m_p)$ such that

$$\nabla^2 f_p(x) = \sum_{i=1}^{m_p} \gamma_{pi}(x)\left\{a_{pi}(x)b_{pi}(x)^T + b_{pi}(x)a_{pi}(x)^T\right\}. \tag{2.3.3}$$

Clearly 2.3.3 is trivially true for $p = 1, \ldots, n$ since by 2.1.1, $\nabla^2 f_p(x) = 0$ $(p = 1, \ldots, n)$. Suppose that 2.3.3 holds for $p = 1, \ldots, \hat{p}$, where $\hat{p} \geq n$. If

$$f_{\hat{p}+1}(x) = f_q(x) \pm f_r(x) \quad (q, r \leq \hat{p}) \tag{2.3.4}$$

then

$$\nabla^2 f_{\hat{p}+1}(x) = \nabla^2 f_q(x) \pm \nabla^2 f_r(x)$$

$$= \sum_{i=1}^{m_q} \gamma_{qi}(x)\left\{a_{qi}(x)b_{qi}(x)^T + b_{qi}(x)a_{qi}(x)^T\right\} \pm$$

$$\sum_{i=1}^{m_r} \gamma_{ri}(x)\left\{a_{ri}(x)b_{ri}(x)^T + b_{ri}(x)a_{ri}(x)^T\right\}$$

$$= \sum_{i=1}^{m_s} \gamma_{si}(x)\left\{a_{si}(x)b_{si}(x)^T + b_{si}(x)a_{si}(x)^T\right\},$$

where $s = \hat{p} + 1$,

$$m_s = m_q + m_r, \tag{2.3.5}$$

$$\gamma_{si}(x) = \begin{cases} \gamma_{qi}(x) & (i = 1, \ldots, m_q), \\ \pm\gamma_{r,i-m_q}(x) & (i = m_q + 1, \ldots, m_s), \end{cases} \tag{2.3.6}$$

$$a_{si}(x) = \begin{cases} a_{qi}(x) & (i = 1, \ldots, m_q), \\ a_{r,i-m_q}(x) & (i = m_q + 1, \ldots, m_s), \end{cases} \tag{2.3.7}$$

and

$$b_{si}(x) = \begin{cases} b_{qi}(x) & (i = 1, \ldots, m_q), \\ b_{r,i-m_q}(x) & (i = m_q + 1, \ldots, m_s). \end{cases} \tag{2.3.8}$$

If

$$f_{\hat{p}+1}(x) = f_q(x)f_r(x) \quad (q, r \leq \hat{p}) \tag{2.3.9}$$

then

$$\nabla^2 f_{\hat{p}+1}(x) = \left\{\nabla^2 f_q(x)\right\} f_r(x) + f_q(x)\left\{\nabla^2 f_r(x)\right\} +$$

$$\nabla f_q(x)\nabla f_r(x)^T + \nabla f_r(x)\nabla f_q(x)^T$$

$$= \sum_{i=1}^{m_q} \gamma_{qi}(x)\left\{a_{qi}(x)b_{qi}(x)^T + b_{qi}(x)a_{qi}(x)^T\right\} f_r(x) +$$

$$\sum_{i=1}^{m_r} \gamma_{ri}(x)\left\{a_{ri}(x)b_{ri}(x)^T + b_{ri}(x)a_{ri}(x)^T\right\} f_q(x) +$$

$$\left\{\nabla f_q(x)\nabla f_r(x)^T + \nabla f_r(x)\nabla f_q(x)^T\right\}$$

$$= \sum_{i=1}^{m_s} \gamma_{si}(x)\left\{a_{si}(x)b_{si}(x)^T + b_{si}(x)a_{si}(x)^T\right\},$$

where $s = \hat{p} + 1$,

$$m_s = m_q + m_r + 1, \qquad 2.3.10$$

$$\gamma_{si}(x) = \begin{cases} \gamma_{qi}(x) f_r(x) & (i = 1, \ldots, m_q), \\ \gamma_{r,i-m_q}(x) f_q(x) & (i = m_q + 1, \ldots, m_q + m_r), \\ 1 & (i = m_s), \end{cases} \qquad 2.3.11$$

$$a_{si}(x) = \begin{cases} a_{qi}(x) & (i = 1, \ldots, m_q), \\ a_{r,i-m_q}(x) & (i = m_q + 1, \ldots, m_q + m_r), \\ \nabla f_q(x) & (i = m_s), \end{cases} \qquad 2.3.12$$

and

$$b_{si}(x) = \begin{cases} b_{qi}(x) & (i = 1, \ldots, m_q), \\ b_{r,i-m_q}(x) & (i = m_q + 1, \ldots, m_q + m_r), \\ \nabla f_r(x) & (i = m_s). \end{cases} \qquad 2.3.13$$

If

$$f_{\hat{p}+1}(x) = f_q(x)/f_r(x) \quad (q, r \leq \hat{p}), \qquad 2.3.14$$

then

$$\nabla^2 f_{\hat{p}+1}(x) = \frac{1}{f_r(x)} \nabla^2 f_q(x) - \frac{f_q(x)}{\{f_r(x)\}^2} \nabla^2 f_r(x) -$$

$$\frac{1}{\{f_r(x)\}^2} \left\{ \nabla f_r(x) \nabla f_q(x)^T + \nabla f_q(x) \nabla f_r(x)^T \right\} +$$

$$\frac{2 f_q(x)}{\{f_r(x)\}^3} \nabla f_r(x) \nabla f_r(x)^T$$

$$= \sum_{i=1}^{m_q} \frac{\gamma_{qi}(x)}{f_r(x)} \left\{ a_{qi}(x) b_{qi}(x)^T + b_{qi}(x) a_{qi}(x)^T \right\} -$$

$$\sum_{i=1}^{m_r} \frac{f_q(x) \gamma_{ri}(x)}{\{f_r(x)\}^2} \left\{ a_{ri}(x) b_{ri}(x)^T + b_{ri}(x) a_{ri}(x)^T \right\} -$$

$$\frac{1}{\{f_r(x)\}^2} \left\{ \nabla f_r(x) \nabla f_q(x)^T + \nabla f_q(x) \nabla f_r(x)^T \right\} +$$

$$\frac{2 f_q(x)}{\{f_r(x)\}^3} \nabla f_r(x) \nabla f_r(x)^T$$

$$= \sum_{i=1}^{m_s} \gamma_{si}(x) \left\{ a_{si}(x) b_{si}(x)^T + b_{si}(x) a_{si}(x)^T \right\},$$

where $s = \hat{p} + 1$,

$$m_s = m_q + m_r + 2, \tag{2.3.15}$$

$$\gamma_{si}(x) = \begin{cases} \gamma_{qi}(x)/f_r(x) & (i = 1, \ldots, m_q), \\ -f_q(x)\gamma_{r,i-m_q}(x)/\{f_r(x)\}^2 & (i = m_q + 1, \ldots, m_q + m_r), \\ -1/\{f_r(x)\}^2 & (i = m_q + m_r + 1), \\ f_q(x)/\{f_r(x)\}^3 & (i = m_s), \end{cases} \tag{2.3.16}$$

$$a_{si}(x) = \begin{cases} a_{qi}(x) & (i = 1, \ldots, m_q), \\ a_{r,i-m_q}(x) & (i = m_q + 1, \ldots, m_q + m_r), \\ \nabla f_r(x) & (i = m_q + m_r + 1), \\ \nabla f_r(x) & (i = m_s), \end{cases} \tag{2.3.17}$$

and

$$b_{si}(x) = \begin{cases} b_{qi}(x) & (i = 1, \ldots, m_q), \\ b_{r,i-m_q}(x) & (i = m_q + 1, \ldots, m_q + m_r), \\ \nabla f_q(x) & (i = m_q + m_r + 1), \\ \nabla f_r(x) & (i = m_s). \end{cases} \tag{2.3.18}$$

Finally, if

$$f_{\hat{p}+1}(x) = T(f_q(x)) \quad (q \leq \hat{p}), \tag{2.3.19}$$

where $T \in \mathcal{F}$, then

$$\nabla^2 f_{\hat{p}+1}(x) = \ddot{T}(f_q(x))\nabla f_q(x)\nabla f_q(x)^T + \dot{T}(f_q(x))\nabla^2 f_q(x),$$

where $\dot{T}(u) = dT(u)/du$ and $\ddot{T}(u) = d^2T(u)/du^2$, so

$$\nabla^2 f_{\hat{p}+1}(x) = \sum_{i=1}^{m_q} \gamma_{qi}(x)\dot{T}(f_q(x))\left\{a_{qi}(x)b_{qi}(x)^T + b_{qi}(x)a_{qi}(x)^T\right\} +$$

$$\tfrac{1}{2}\ddot{T}(f_q(x))\left\{\nabla f_q(x)\nabla f_q(x)^T + \nabla f_q(x)\nabla f_q(x)^T\right\}$$

$$= \sum_{i=1}^{m_s} \gamma_{si}(x)\left\{a_{si}(x)b_{si}(x)^T + b_{si}(x)a_{si}(x)^T\right\},$$

where $s = \hat{p} + 1$,

$$m_s = m_q + 1, \tag{2.3.20}$$

$$\gamma_{si}(x) = \begin{cases} \gamma_{qi}(x)\dot{T}(f_q(x)) & (i = 1, \ldots, m_q), \\ \\ \tfrac{1}{2}\ddot{T}(f_q(x)) & (i = m_q), \end{cases} \tag{2.3.21}$$

$$a_{si}(x) = \begin{cases} a_{qi}(x) & (i = 1, \ldots, m_q), \\ \\ \nabla f_q(x) & (i = m_s), \end{cases} \tag{2.3.22}$$

and

$$b_{si}(x) = \begin{cases} b_{qi}(x) & (i = 1, \ldots, m_q), \\ \\ \nabla f_q(x) & (i = m_s). \end{cases} \qquad 2.3.23$$

Therefore 2.3.3 holds for $\hat{p}+1$ if it holds for $\hat{p}$. Therefore by finite induction on p, 2.3.2 holds. □

Theorem 2.3.2

If $F : R^n \to R^n$ is a differentiable function and $J : R^n \to R^{n \times n}$ is defined by 2.3.1 then $\exists a_i : R^n \to R^n$ $(i = 1, \ldots, n)$ such that

$$J(x)^T J(x) = \sum_{i=1}^{n} a_i(x) a_i(x)^T. \qquad 2.3.24$$

Proof

For $i, j = 1, \ldots, n$,

$$\begin{aligned} \big(J(x)^T J(x)\big)_{ij} &= \sum_{k=1}^{n} \big(J(x)^T\big)_{ik} \big(J(x)\big)_{kj} \\ &= \sum_{k=1}^{n} \partial_i F_k(x) \partial_j F_k(x) \\ &= \sum_{k=1}^{n} \big(\nabla F_k(x) \nabla F_k(x)^T\big)_{ij}. \end{aligned}$$

So

$$J(x)^T J(x) = \sum_{k=1}^{n} \nabla F_k(x) \nabla F_k(x)^T.$$

Therefore 2.3.24 holds with

$$a_i(x) = \nabla F_i(x) \quad (i = 1, \ldots, n). \square \tag{2.3.25}$$

The proofs of Theorems 2.3.1 and 2.3.2 are constructive in that Theorem 2.3.1 contains an algorithm for determining $\gamma_i(x)$, $a_i(x)$, and $b_i(x)$. In Theorem 2.3.2, the $a_i(x)$ in 2.3.24 can be determined from 2.3.25.

In order to determine the $\gamma_{si}(x)$, $a_{si}(x)$, and $b_{si}(x)$ from 2.3.4–2.3.23, it is necessary to determine $\nabla f_p(x)$ $(p \geq 1)$. If

$$f_p(x) = x_p \quad (1 \leq p \leq n) \tag{2.3.26}$$

then

$$\nabla f_p(x) = e_p, \tag{2.3.27}$$

where e_p is column p of the $n \times n$ unit matrix. If $p > n$ and

$$f_p(x) = f_q(x) \pm f_r(x) \quad (q, r < p) \tag{2.3.28}$$

then for $i = 1, \ldots, n$,

$$\partial_i f_p(x) = \partial_i f_q(x) \pm \partial_i f_r(x),$$

whence

$$\nabla f_p(x) = \nabla f_q(x) \pm \nabla f_r(x). \tag{2.3.29}$$

If $p > n$ and

$$f_p(x) = f_q(x) f_r(x) \quad (q, r < p) \tag{2.3.30}$$

then for $i = 1, \ldots, n$,

$$\partial_i f_p(x) = \{\partial_i f_q(x)\} f_r(x) + f_q(x) \{\partial_i f_r(x)\},$$

whence

$$\nabla f_p(x) = f_r(x) \nabla f_q(x) + f_q(x) \nabla f_r(x). \tag{2.3.31}$$

If $p > n$ and

$$f_p(x) = f_q(x) / f_r(x) \quad (q, r < p) \tag{2.3.32}$$

then for $i = 1, \ldots, n$,

$$\partial_i f_p(x) = \{\partial_i f_q(x) - f_q(x) \partial_i f_r(x) / f_r(x)\} / f_r(x)$$

whence

$$\nabla f_p(x) = \frac{1}{f_r(x)} \nabla f_q(x) - \frac{f_q(x)}{\{f_r(x)\}^2} \nabla f_r(x). \tag{2.3.33}$$

Finally, if $p > n$ and

$$f_p(x) = T(f_q(x)) \quad (q < p), \tag{2.3.34}$$

where $T \in \mathcal{F}$, then for $i = 1, \ldots, n$,

$$\partial_i f_p(x) = \dot{T}(f_q(x)) \partial_i f_q(x)$$

whence

$$\nabla f_p(x) = \dot{T}(f_q(x))\nabla f_q(x). \qquad 2.3.35$$

Thus if $\left(f_p(x)\right)$ is a sequence of CFFs then for $p \geq 1$, $\nabla f_p(x)$ can be determined recursively from 2.3.26–2.3.35. The $\gamma_i(x)$, $a_i(x)$, and $b_i(x)$ may be determined recursively as follows. For $p = 1, \ldots, n$,

$$\nabla^2 f_p(x) = \left(0\right)_{n\times n},$$

so

$$m_p = 1, \quad \gamma_1(x) = 1, \quad a_1(x) = \left(0\right)_{n\times 1}, \quad \text{and} \quad b_1(x) = \left(0\right)_{n\times 1}.$$

If $p = n + 1$ and

$$f_p(x) = f_q(x) \pm f_r(x) \quad (q, r \leq n)$$

then

$$\nabla^2 f_p(x) = \left(0\right)_{n\times n},$$

so

$$m_p = 1, \quad \gamma_1(x) = 1, \quad a_1(x) = \left(0\right)_{n\times 1}, \quad \text{and} \quad b_1(x) = \left(0\right)_{n\times 1}.$$

If $p = n + 1$ and

$$f_p(x) = f_q(x) f_r(x) \quad (q, r \leq n)$$

then

$$\nabla^2 f_p(x) = \nabla f_q(x)\nabla f_r(x)^T + \nabla f_r(x)\nabla f_q(x)^T$$

$$= e_q e_r^T + e_r e_q^T,$$

so

$$m_p = 1, \quad \gamma_1(x) = 1, \quad a_1(x) = e_q, \quad \text{and} \quad b_1(x) = e_r.$$

If $p = n + 1$ and

$$f_p(x) = f_q(x)/f_r(x) \quad (q, r \leq n)$$

then

$$\nabla^2 f_p(x) = -\frac{1}{\{f_r(x)\}^2}\left\{\nabla f_r(x)\nabla f_q(x)^T + \nabla f_q(x)\nabla f_r(x)^T\right\} +$$

$$\frac{2f_q(x)}{\{f_r(x)\}^3}\nabla f_r(x)\nabla f_r(x)^T$$

$$= -\frac{1}{x_r^2}\left(e_r e_q^T + e_q e_r^T\right) + \frac{2x_q}{x_r^3} e_r e_r^T,$$

so

$$m_p = 2,$$

$$\gamma_1(x) = -1/x_r^2, \quad a_1(x) = e_r, \quad b_1(x) = e_q,$$

and

$$\gamma_2(x) = x_q/x_r^3, \quad a_2(x) = e_r, \quad b_2(x) = e_r.$$

If $p = n + 1$ and

$$f_p(x) = T(f_q(x)) \quad (q \leq n)$$

then

$$\nabla^2 f_p(x) = \ddot{T}(f_q(x))\nabla f_q(x)\nabla f_q(x)^T,$$

so

$$m_p = 1, \quad \gamma_1(x) = \tfrac{1}{2}\ddot{T}(x_q), \quad a_1(x) = e_q, \quad \text{and} \quad b_1(x) = e_q.$$

For $p > n+1$, m_p, $\gamma_{pi}(x)$, $a_{pi}(x)$, and $b_{pi}(x)$, $(i = 1, \ldots, m_p)$ can then be determined recursively from 2.3.4–2.3.23 with $s = p$.

The formulae which have been derived in this section have been used [SheW--85b] to construct the $\mathcal{ALGLIB}$ procedure *compute.sisser.functions*. If the CFF $f : R^3 \rightarrow R^1$ is defined by

$$f(x) = cos(x_1 + x_2 x_3), \qquad 2.3.36$$

and f is a pointer to the $\mathcal{ALGLIB}$ data-structure corresponding to $f : R^3 \rightarrow R^1$, then the statement

$$\underline{let}\ S = compute.sisser.functions(f)$$

gives a pointer S to a data-structure with field names ***sisser.m***, ***sisser.gamma***, ***sisser.a***, and ***sisser.b***, so that the statements

$$\underline{let}\ m = S(sisser.m)$$

$$\underline{let}\ gamma = S(sisser.gamma)$$

$$\underline{let}\ a = S(sisser.a)$$

$$\underline{let}\ b = S(sisser.b)$$

determine $m \geq 1$, $\gamma : R^n \to R^m$, $a : R^n \to R^{m\times n}$, and $b : R^n \to R^{m\times n}$ such that 2.3.2 holds. If

$$f_i(x) = x_i \quad (i = 1, 2, 3),$$

$$f_4(x) = f_2(x) f_3(x),$$

$$f_5(x) = f_1(x) + f_4(x),$$

and

$$f_6(x) = cos\{f_5(x)\},$$

then $f(x) = f_6(x)$, and using the preceding formulae one obtains

$$\nabla^2 f(x) = \sum_{i=1}^{2} \gamma_i(x) \left\{ a_i(x) b_i(x)^T + b_i(x) a_i(x)^T \right\},$$

where

$$\gamma_1(x) = -\tfrac{1}{2} cos\{f_5(x)\} = -\tfrac{1}{2} cos(x_1 + x_2 x_3),$$

$$\gamma_2(x) = -\, sin\{f_5(x)\} = -sin(x_1 + x_2 x_3),$$

$$a_1(x) = (1, f_3(x), f_2(x))^T = (1, x_3, x_2)^T,$$

$$b_1(x) = (1, f_3(x), f_2(x))^T = (1, x_3, x_2)^T,$$

$$a_2(x) = (0, 0, 1)^T,$$

and

$$b_2(x) = (0, 1, 0)^T.$$

Pointers to the expressions $\gamma_i(x)$, $a_{ij}(x)$, $b_{ij}(x)$ $(i = 1, 2; j = 1, 2, 3)$ are $gamma(i)$,

$a(i,j)$, and $b(i,j)$ $(i = 1,2; j = 1,2,3)$, so one could exhibit the expressions $\gamma_i(x)$, $a_{ij}(x)$, and $b_{ij}(x)$ by using the $\mathcal{ALGLIB}$ procedure *function.format* as follows.

```
for i = 1 to m do
    write "'n", function.format(gamma(i))
for i = 1 to m do
    for j = 1 to n do
        write "'n", function.format(a(i, j))
        write "'n", function.format(b(i, j))
```

where $m = 2$ and $n = 3$.

It is convenient, for use in Chapter 3, to express $\nabla^2 f(x)$ in the form

$$\nabla^2 f(x) = D + \sum_{i=1}^{r} s_i(x) q_i(x) q_i(x)^T, \qquad 2.3.37$$

where $D = \eta I$ $(\eta > 0)$, $r = 2m + n$, and for $i = 2m + 1, \ldots, r$,

$$s_i(x) = -\eta, \qquad 2.3.38$$

and

$$q_i(x) = e_{i-2m}. \qquad 2.3.39$$

The Hessian $G(x) = \nabla^2 f(x)$ may be expressed in the form 2.3.37 after having been expressed in the form 2.3.2 as follows.

Suppressing the argument x for brevity, we have

$$G = \sum_{i=1}^{m} \gamma_i \left\{ a_i b_i^T + b_i a_i^T \right\}$$

$$= \sum_{i=1}^{m} \left\{ (\tfrac{\gamma_i}{2})(a_i + b_i)(a_i + b_i)^T + (-\tfrac{\gamma_i}{2})(a_i - b_i)(a_i - b_i)^T \right\}. \qquad 2.3.40$$

Let

$$s_i = \begin{cases} \frac{1}{2}\gamma_i & (i = 1, \ldots, m), \\ -\frac{1}{2}\gamma_{i-m} & (i = m+1, \ldots, 2m), \end{cases}$$

and

$$q_i = \begin{cases} a_i + b_i & (i = 1, \ldots, m), \\ a_{i-m} - b_{i-m} & (i = m+1, \ldots, 2m). \end{cases}$$

Then

$$G = \sum_{i=1}^{2m} s_i q_i q_i^T.$$

Now

$$D = \eta I$$

$$= \eta \sum_{i=1}^{n} e_i e_i^T,$$

where I is the $n \times n$ unit matrix and e_i $(i = 1, \ldots, n)$ are its columns. So if, for $i = 2m+1, \ldots, r$, s_i and q_i are given by 2.3.38 and 2.3.39 respectively, then 2.3.37 holds.

Using the *ALGLIB* procedures ***compute.sisser.functions***, ***function.op.real***, ***function.op.function***, ***op.function***, and ***string.to.function***, it is easy to write an S-algol procedure which returns a pointer to a structure containing m and pointers to the CFFs $s_i : R^n \to R^1$ and $q_i : R^n \to R^n$ $(i = 1, \ldots, r)$.

Finally, if $J : R^n \to R^{n \times n}$ is defined by 2.3.1 and $A : R^n \to R^{n \times n}$ is defined by

$$A = J^T J,$$

where the argument x is omitted for brevity, then for $i, j = 1, \ldots, n$,

$$\begin{aligned} A_{ij} &= \sum_{k=1}^{n} \left(J^T\right)_{ik} \left(J\right)_{kj} \\ &= \sum_{k=1}^{n} \left(\partial_i F_k\right) \left(\partial_j F_k\right) \\ &= \sum_{k=1}^{n} \left(\nabla F_k \nabla F_k^T\right)_{ij}, \end{aligned}$$

so if

$$q_k(x) = \nabla F_k(x) \quad (k = 1, \ldots, n),$$

then

$$\begin{aligned} A(x) &= J(x)^T J(x) \\ &= \sum_{k=1}^{n} q_k(x) q_k(x)^T. \end{aligned} \qquad 2.3.41$$

CHAPTER 3

Modifications of Sisser's Method

3.1 Newton's Method

The most frequently-used method for minimizing the objective function $f : R^n \to R^1$ without constraints when $f \in C^2(R^n)$ is Newton's method or some modification thereof. The basic idea behind Newton's method is that f is approximated locally with a quadratic function which is obtained by using Taylor's theorem to expand f about the current estimate of an unconstrained minimizer x^* of f up to and including second order terms and then the quadratic function is minimized exactly to obtain a new estimate of x^*. More explicitly, if $x^{(k)} \in R^n$ $(k \geq 0)$ is the current estimate of x^* then f is approximated near $x^{(k)}$ using the quadratic function $q : R^n \to R^1$ defined by

$$q(x) = f(x^{(k)}) + g(x^{(k)})^T(x - x^{(k)}) + \tfrac{1}{2}(x - x^{(k)})^T G(x^{(k)})(x - x^{(k)}), \qquad 3.1.1$$

where $g(x^{(k)}) = \nabla f(x^{(k)})$ and $G(x^{(k)}) = \nabla^2 f(x^{(k)})$. The function q has the unique minimizer $x^{(k+1)}$ given by

$$x^{(k+1)} = x^{(k)} - G(x^{(k)})^{-1} g(x^{(k)}) \qquad 3.1.2$$

if $G(x^{(k)})$ is positive definite.

If $x^{(k)}$ is not a good estimate of x^* then q may be a very poor approximation to f near $x^{(k)}$ and $x^{(k+1)}$ may be a poorer estimate of x^* than is $x^{(k)}$, leading possibly to divergence of the sequence $\left(x^{(k)}\right)$ rather than the hoped-for convergence to x^*.

If $G(x^{(k)})$ is not positive definite but is non-singular then the Newton step $x^{(k+1)} - x^{(k)}$, where $x^{(k+1)}$ is given by 3.1.2, may be towards a saddle point of f, while if $G(x^{(k)})$ is singular then $x^{(k+1)}$ is not defined.

In practice it is necessary to replace $G^{(k)} = G(x^{(k)})$ with a positive definite matrix $\tilde{G}^{(k)}$ of the form

$$\tilde{G}^{(k)} = G^{(k)} + \mu^{(k)} I, \qquad 3.1.3$$

where $\mu^{(k)} > 0$ and I is the $n \times n$ unit matrix, and to introduce a steplength parameter $\lambda^{(k)}$ which satisfies certain conditions [OrtR--70] [DenS--83] into 3.1.2 to obtain

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)} \tilde{G}^{(k)-1} g^{(k)}, \qquad 3.1.4$$

where $g^{(k)} = g(x^{(k)})$, and $\tilde{G}^{(k)}$ is given by 3.1.3.

A very effective algorithm for determining $\lambda^{(k)}$ has been described by Dennis and Schnabel [DenS--83] and is used to obtain the numerical results which are reported subsequently in this thesis.

In this chapter, we use notations $M(R^n)$, $I(M(R^n))$ and $I(R^n)$ to denote the sets of real $n \times n$ matrices, of real $n \times n$ interval matrices and of real $n \times 1$ interval vectors respectively.

The remainder of this chapter is organized as follows. Section 3.2 contains a description of Sisser's modifications [Sis---82a] of Newton's method, and Sections 3.3 – 3.7 contain 12 new minor modifications of Sisser's algorithm S which is described in §3.2. Sections 3.8 and 3.9 contain modifications of Newton's method in which $\mathcal{ALGLIB}$ is used to construct a modified $n \times n$ Hessian matrix which is positive definite. Section 3.10 contains the sum-

mary of Sisser's methods and the modifications which are described in §§3.2–3.7. Section 3.11 contains a flow diagram for the minimization algorithms. Section 3.12 contains numerical results for the methods which are described in Sections 3.2–3.9. Finally, future work concerning applications of $\mathcal{ALGLIB}$ is presented in Section 3.13.

3.2 Sisser's Method

Sisser [Sis---82a] has described some modifications of Newton's method in which, given an intial estimate $x^{(0)} \in R^n$ of an unconstrained strong local minimizer x^* of the objective function $f : R^n \to R^1$, the sequence $\left(x^{(k)}\right)$ is generated from

$$x^{(k+1)} = x^{(k)} - \lambda^{(k)}\left(G^{(k)} + \mu^{(k)}I\right)^{-1} g^{(k)} \quad (k \geq 0), \tag{3.2.1}$$

where the $\lambda^{(k)}$ are determined so as to satisfy

$$f^{(k+1)} \leq f^{(k)} + \alpha\lambda^{(k)} p^{(k)^T} g^{(k)} \tag{3.2.2}$$

and

$$g^{(k+1)^T} p^{(k)} \geq \beta p^{(k)^T} g^{(k)}, \tag{3.2.3}$$

where $f^{(k)} = f(x^{(k)})$, $g^{(k)} = g(x^{(k)}) = \nabla f(x^{(k)})$, $G^{(k)} = G(x^{(k)}) = \nabla^2 f(x^{(k)})$,

$$p^{(k)} = -\left(G^{(k)} + \mu^{(k)}I\right)^{-1} g^{(k)}, \tag{3.2.4}$$

$0 < \alpha < \beta < 1$, and $\alpha < \frac{1}{2}$.

The $\mu^{(k)}$ are determined by expressing $G^{(k)}$ as a sum of dyads and then using the Sherman Morrison Woodbury (SMW) formula [SheM--49] [Woo---50], as explained subsequently.

Theorem 3.2.1

Let $Q \in R^{n\times n}$ be nonsingular and let $u, v \in R^n$. Then $(Q+uv^T)^{-1}$ exists if and only if $v^TQ^{-1}u+1 \neq 0$. Furthermore, if $v^TQ^{-1}u+1 \neq 0$ then

$$\left(Q+uv^T\right)^{-1} = Q^{-1} - \left\{1/(1+v^TQ^{-1}u)\right\}Q^{-1}uv^TQ^{-1}. \qquad 3.2.5$$

Proof

Suppose that $v^TQ^{-1}u + 1 = 0$. Then $(Q + uv^T)Q^{-1}u = 0$. Now $u \neq 0$ because $v^TQ^{-1}u + 1 = 0$, so $Q^{-1}u \neq 0$, whence $(Q+uv^T)w = 0$ where $w = Q^{-1}u$. So $Q+uv^T$ has a zero eigenvalue, whence $Q + uv^T$ is singular. Conversely, if $v^TQ^{-1}u + 1 \neq 0$ then

$$(Q+uv^T)\left[Q^{-1} - \left\{1/(1+v^TQ^{-1}u)\right\}Q^{-1}uv^TQ^{-1}\right] = I,$$

whence 3.2.5 holds. □

If, in 3.2.5, $u = sq$ and $v = q$ then

$$\left(Q+sqq^T\right)^{-1} = Q^{-1} - \left\{s/(1+sq^TQ^{-1}q)\right\}Q^{-1}qq^TQ^{-1} \qquad 3.2.6$$

which is the form of the SMW formula which is required in Sisser's modifications of Newton's method. The Hessian $G = G^{(k)}$ is expressed in the form

$$G = D + \sum_{i=1}^{r} s_iq_iq_i^T, \qquad 3.2.7$$

where $D = \eta I$ ($\eta > 0$) as explained in §2.3, and η is determined experimentally as explained in §3.12. The dyads $s_i q_i q_i^T$ are ordered so that for $i = 1, \ldots, r-1$,

$$s_i q_i^T q_i \geq s_{i+1} q_{i+1}^T q_{i+1}. \qquad 3.2.8$$

The Hessian G is inverted by using 3.2.6. Initially $Q = D$ so Q is positive definite, and the r dyads $s_i q_i q_i^T$ are added, one at a time, by using 3.2.6 recursively, in such a way as to preserve positive definiteness. The following results are needed in order to understand how this can be done.

Theorem 3.2.2

If $u, v \in R^n$ then

$$det(I + uv^T) = 1 + v^T u.$$

Proof

Let $U \in R^{n \times n}$ be defined by

$$U = \left(u/(u^T u)^{\frac{1}{2}} \vdots 0 \vdots \cdots \vdots 0\right)_{n \times n}.$$

Then

$$U^T u v^T U = diag(v^T u, 0, \ldots, 0),$$

so that eigenvalues of uv^T are $v^T u$ and 0, the latter with multiplicity $n-1$. Therefore the eigenvalues of $I + uv^T$ are $1 + v^T u$ and 1, the latter with multiplicity $n-1$. Therefore $det(I + uv^T) = 1 + v^T u$. □

Theorem 3.2.3

Let $Q \in R^{n\times n}$ be symmetric with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$. Let $u \in R^n$, and let the eigenvalues of $Q + uu^T$ be $\mu_1 \geq \ldots \geq \mu_n$. Then $\mu_1 \geq \lambda_1 \geq \mu_2 \geq \ldots \geq \mu_n \geq \lambda_n$. Furthermore, if the eigenvalues of $Q - uu^T$ are $\nu_1 \geq \ldots \geq \nu_n$ then $\lambda_1 \geq \nu_1 \geq \ldots \geq \lambda_n \geq \nu_n$.

Proof

See [Wol---78] Appendix I. □

Theorem 3.2.4

If $Q \in R^{n\times n}$ is symmetric positive definite, $s \in R$, and $q \in R^n$, then $Q + sqq^T$ is positive definite if and only if $1 + sq^TQ^{-1}q > 0$.

Proof

By Theorem 3.2.2,

$$det(Q + sqq^T) = det\left(Q(I + sQ^{-1}qq^T)\right)$$

$$= det(Q)det(I + (Q^{-1}q)(sq)^T)$$

$$= det(Q)(1 + sq^TQ^{-1}q). \qquad 3.2.9$$

Suppose that $1 + sq^TQ^{-1}q > 0$. Then by 3.2.9,

$$det(Q+sqq^T) > 0, \qquad 3.2.10$$

since $det(Q) > 0$ because Q is positive definite. Now if the eigenvalues of Q are $\lambda_1 \geq \ldots \geq \lambda_n$ and the eigenvalues of $Q + sqq^T$ are $\mu_1 \geq \ldots \geq \mu_n$ then by Theorem 3.2.3 with $u = |s|^{\frac{1}{2}}q$,

$\mu_1 \geq \lambda_1 \geq \ldots \geq \mu_n \geq \lambda_n$ and if the eigenvalues of $Q - sqq^T$ are $\nu_1 \geq \ldots \geq \nu_n$ then $\lambda_1 \geq \nu_1 \geq \ldots \geq \lambda_n \geq \nu_n$. Also

$$det(Q + sqq^T) = \prod_{i=1}^{n} \mu_i,$$

$$det(Q - sqq^T) = \prod_{i=1}^{n} \nu_i,$$

and

$$det(Q) = \prod_{i=1}^{n} \lambda_i.$$

If Q is positive definite then $\lambda_n > 0$ so $\mu_n > 0$ and $\nu_{n-1} > 0$. Only ν_n can be negative. So at most one eigenvalue of $Q + sqq^T$ can be negative. So if $det(Q + sqq^T) > 0$ then every eigenvalue of $Q + sqq^T$ is positive whence $Q + sqq^T$ is positive definite.

Conversely, if $1 + sq^T Q^{-1} q \leq 0$ then by 3.2.9 $det(Q + sqq^T) \leq 0$ whence $Q + sqq^T$ has a negative eigenvalue. Therefore $Q + sqq^T$ is not positive definite. □

Let

$$H^{(0)} = D^{-1}$$

$$= (1/\eta) I, \qquad 3.2.11$$

where D is as in 3.2.7, and let the sequence $\left(H^{(j)}\right)$ be generated from 3.2.6 according to

$$H^{(j)} = H^{(j-1)} - \left\{ s_j / (1 + s_j q_j^T H^{(j-1)} q_j) \right\} H^{(j-1)} q_j q_j^T H^{(j-1)} \quad (1 \leq j \leq r), \qquad 3.2.12$$

so that

$$H^{(j)} = \left(D + \sum_{i=1}^{j} s_i q_i q_i^T\right)^{-1} \quad (1 \leq j \leq r), \qquad 3.2.13$$

provided that $H^{(j)}$ exists. Suppose that for some $j \geq 1$, $D + \sum_{i=1}^{j-1} s_i q_i q_i^T$ is positive definite. Then $H^{(j-1)}$ exists and is positive definite. Therefore by Theorem 3.2.4, if $s_j > 0$ then $D + \sum_{i=1}^{j} s_i q_i q_i^T$ is positive definite because $1 + s_j q_j^T H^{(j-1)} q_j > 0$. Therefore $H^{(j)}$ exists and is positive definite.

Now by 3.2.8 the dyads in 3.2.7 are ordered so that those with positive coefficients come first. Therefore for some $k \leq 2m$, where m is as in § 2.3, $s_i > 0$ $(i = 1, \ldots, k)$ and $D + \sum_{i=1}^{k} s_i q_i q_i^T$ is positive definite. If, after all the dyads have been added, positive definiteness has been retained, so that $H^{(j)}$ exists and is positive definite for $j = 0, \ldots, r$ then $G^{(k)^{-1}} = G^{-1} = H^{(r)}$ and $x^{(k+1)}$ is computed from 3.2.1 with $\mu^{(k)} = 0$. If, however, it is found that dyads can no longer be added without causing $D + \sum_{i=1}^{l} s_i q_i q_i^T$ to lose positive definiteness for some $l \leq r$, then a lower bound $\mu^{(k)}$ on the smallest eigenvalue of $G^{(k)}$ is computed and is used in 3.2.1. The following results are needed in order to understand how this can be done.

Theorem 3.2.5

Let $P, Q \in R^{n \times n}$ be symmetric with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ and $\mu_1 \geq \ldots \geq \mu_n$ respectively. Let $R = P + Q$ and suppose that R has eigenvalues $\nu_1 \geq \ldots \geq \nu_n$. Then

$$\nu_i \geq \lambda_i + \mu_n \qquad (i = 1, \ldots, n),$$

and

$$\nu_i \leq \lambda_i + \mu_1 \qquad (i = 1 \ldots, n).$$

Proof

See [Wol---78] Appendix I. □

Theorem 3.2.6

Let $P \in R^{n\times n}$ be symmetric, let $q \in R^n$, and let $s < 0$. If λ_n is the smallest eigenvalue of P and ν_n is the smallest eigenvalue of R where

$$R = P + sqq^T,$$

then

$$\nu_n \geq \lambda_n + sq^Tq.$$

Proof

By Theorem 3.2.5 with $Q = sqq^T$, $\mu_n = sq^Tq$, $\mu_i = 0$ $(i = 1, \ldots, n-1)$,

$$\nu_i \geq \lambda_i + sq^Tq \quad (i = 1, \ldots, n),$$

where $\lambda_1 \geq \ldots \geq \lambda_n$ and $\nu_1 \geq \ldots \geq \nu_n$ are the eigenvalues of P and R respectively. □

Theorem 3.2.7

Let $P \in R^{n\times n}$ be symmetric positive definite, let $q_i \in R^n$ $(i = 1, \ldots, r)$ and let $c_i < 0$ $(i = 1, \ldots, r)$. If

$$R = P + \sum_{i=1}^{r} c_i q_i q_i^T$$

has eigenvalues $\nu_1 \geq \ldots \geq \nu_n$ then

$$\nu_n \geq \sum_{i=1}^{r} c_i q_i^T q_i.$$

Proof

Let the eigenvalues of P be $\lambda_1 \geq \ldots \geq \lambda_n$, let

$$R^{(t)} = P + \sum_{i=1}^{t} c_i q_i q_i^T \quad (t = 1, \ldots, r),$$

and let the eigenvalues of $R^{(t)}$ be $\nu_1^{(t)} \geq \ldots \geq \nu_n^{(t)}$. Then by Theorem 3.2.6,

$$\nu_n^{(1)} \geq \lambda_n + c_1 q_1^T q_1.$$

But P is positive definite so $\lambda_n > 0$ whence $\nu_n^{(1)} \geq c_1 q_1^T q_1$. Suppose that for some $t \geq 1$,

$$\nu_n^{(t)} \geq \sum_{i=1}^{t} c_i q_i^T q_i.$$

Now

$$R^{(t+1)} = R^{(t)} + c_{t+1} q_{t+1} q_{t+1}^T,$$

so by Theorem 3.2.6,

$$\nu_n^{(t+1)} \geq \nu_n^{(t)} + c_{t+1} q_{t+1}^T q_{t+1}$$

$$\geq \sum_{i=1}^{t+1} c_i q_i^T q_i.$$

So by finite induction on t,

$$\nu_n = \nu_n^{(r)}$$

$$\geq \sum_{i=1}^{r} c_i q_i^T q_i. \square$$

Suppose that, in 3.2.7, the first l dyads have been added to D, one at a time, and that the test for positive definiteness in Theorem 3.2.4 has been passed for each dyad. Then for

$j = 1, \ldots, l \leq r$, $H^{(j)}$ is positive definite, where $H^{(j)}$ is defined by 3.2.13. Let

$$G = D + \sum_{i=1}^{k} s_i q_i q_i^T + \sum_{i=k+1}^{l} s_i q_i q_i^T + \sum_{i=l+1}^{r} s_i q_i q_i^T, \qquad 3.2.14$$

where $s_i > 0$ $(i = 1, \ldots, k)$ and $s_i < 0$ $(i = k+1, \ldots, r)$. Suppose that

$$1 + s_{l+1} q_{l+1}^T H^{(l)} q_{l+1} \leq 0. \qquad 3.2.15$$

Then by Theorem 3.2.7, ν_n, the smallest eigenvalue of G, is bounded below according to

$$\nu_n \geq \sum_{i=l+1}^{r} s_i q_i^T q_i,$$

so if

$$\mu^{(k)} = \sum_{i=l+1}^{r} |s_i| q_i^T q_i \qquad 3.2.16$$

then $G^{(k)} + \mu^{(k)} I$ is positive definite and can be used in 3.2.1 to compute $x^{(k+1)}$.

The use of 3.2.16 is an inexpensive way of determining a suitable value for $\mu^{(k)}$, but as pointed out by Sisser [Sis---82a] a better value of $\mu^{(k)}$ could be obtained by splitting the $(l+1)^{\text{th}}$ and all subsequent dyads and trying to absorb parts of those dyads without losing positive definiteness. Let

$$p_{l+1} = (\delta - 1)/(s_{l+1} q_{l+1}^T H^{(l)} q_{l+1}), \qquad 3.2.17$$

where $\delta > 0$ is determined experimentally as explained in §3.12 and $H^{(l)}$ is given by 3.2.13. Then

$$1 + p_{l+1} s_{l+1} q_{l+1}^T H^{(l)} q_{l+1} = \delta,$$

so by Theorem 3.2.4, the dyad $p_{l+1} s_{l+1} q_{l+1} q_{l+1}^T$ can be absorbed without losing positive definiteness to obtain $H^{(l+1)}$ where $H^{(l+1)}$ is given by 3.2.12 with $j = l + 1$ and $p_{l+1} s_{l+1}$ replacing s_{l+1}, and by 3.2.13

$$H^{(l+1)} = \left(D + \sum_{i=1}^{l} s_i q_i q_i^T + p_{l+1} s_{l+1} q_{l+1} q_{l+1}^T \right)^{-1}.$$

The procedure is repeated for $i = l + 2, \ldots, r$, with

$$p_i = (\delta - 1)/(s_i q_i^T H^{(i-1)} q_i), \qquad 3.2.18$$

in which

$$H^{(i-1)} = \left(D + \sum_{j=1}^{l} s_j q_j q_j^T + \sum_{j=l+1}^{i-1} p_j s_j q_j q_j^T \right)^{-1}.$$

An improved value of $\mu^{(k)}$ for use in 3.2.1 is then given by

$$\mu^{(k)} = \sum_{i=l+1}^{r} (1 - p_i) |s_i| q_i^T q_i. \qquad 3.2.19$$

The matrix $G^{(k)} + \mu^{(k)} I$ is inverted by using 3.2.12 as follows. There are two cases, namely $(i)\ l < n$; $(ii)\ l \geq n$.

$(i)\ l < n$. Set

$$H^{(0)} = \{1/(\eta + \mu^{(k)})\} I$$

and use 3.2.12 to obtain $(G^{(k)} + \mu^{(k)}I)^{-1} = H^{(r)}$. There is no need to test for positive definiteness when adding a dyad, because $\mu^{(k)}$ is large enough to ensure that $G^{(k)} + \mu^{(k)}I$ is positive definite.

(*ii*) $l \geq n$. In this case,

$$H^{(r)} = \left(D + \sum_{i=1}^{l} s_i q_i q_i^T + \sum_{i=l+1}^{r} p_i s_i q_i q_i^T\right)^{-1}$$

has already been computed. The SMW formula is used to add the dyads corresponding to $\mu^{(k)}I$, namely $\mu^{(k)} e_i e_i^T$ $(i = 1, \ldots, n)$ where e_i is column i of the $n \times n$ unit matrix, and then the SMW formula is used to add the $r - l$ dyads $(1 - p_i)s_i q_i q_i^T$ $(i = l + 1, \ldots, r)$ where p_i is given by 3.2.18. Again there is no need to test for positive definiteness.

If, in 3.2.1, $||g^{(k)}||_2 < \varepsilon$ for some k, where $\varepsilon \in R$ is given and is such that $0 < \varepsilon << 1$, and $G^{(k)}$ is positive definite so that $\mu^{(k)} = 0$, then $x^{(k)}$ is accepted as the final estimate of x^*. If however $||g^{(k)}||_2 < \varepsilon$ and $G^{(k)}$ is not at least positive semi-definite then $x^{(k)}$ is a saddle point. In this case, $x^{(k+1)} = x^{(k)} + s^{(k)}$ where $s^{(k)}$ is a small perturbation of random direction such that $||g^{(k+1)}||_2 \geq \varepsilon$.

Sisser's method is expressed as an algorithm in [Sis---82a], and has been implemented in S-algol [ColM--82a] on a VAX-11/785 computer using $\mathcal{ALGLIB}$ [SheW--85] to express the Hessian as a sum of dyads. Sisser's algorithm is referred to in this thesis as Algorithm S (see Figure 3.10.1).

Sisser [Sis---82a] has described 3 modifications of S which are referred to in this thesis as S1, S2, and S3. In S1 (see Figure 3.10.2), $\mu^{(k)}$ is determined from 3.2.16. In S2 (see Figure 3.10.3), if, for some $i \in \{1, \ldots, r\}$,

$$1 + s_i q_i^T H^{(i-1)} q_i \leq 0, \qquad 3.2.20$$

so that by Theorem 3.2.4, positive definiteness would be lost by adding the i^{th} dyad, then in the dyads $i, \ldots, r$, s_j is replaced with $|s_j|$ and $H^{(j)}$ $(j = i, \ldots, r)$ is computed from 3.2.12 to give a positive definite matrix $H^{(r)}$ which is regarded as an approximation to $G^{(k)^{-1}}$.

In S3 (see Figure 3.10.4), if for some $i \in \{1, \ldots, r\}$ 3.2.20 holds then $G^{(k)}$ is replaced with the $n \times n$ unit matrix I, so that a steepest descent step is taken.

3.3 The Modifications M1 and M2 of Sisser's Method

Let $G \in R^{n \times n}$ be given by 3.2.7 and suppose that the dyads have not been ordered so that 3.2.8 holds. Let $H^{(0)}$ be given by 3.2.11 and let the sequence $\left(H^{(j)}\right)$ be generated from 3.2.12. Suppose that for some $i \in \{2, \ldots, 2m\}$,

$$1 + s_j q_j^T H^{(j-1)} q_j > 0 \quad (j = 1, \ldots, i-1),$$

but that

$$1 + s_i q_i^T H^{(i-1)} q_i \leq 0. \qquad 3.3.1$$

Then for $j = 1, \ldots, (i-1)$, $H^{(j)}$ is positive definite, but $H^{(i)}$ would not be positive definite. This suggests that the dyad $s_i q_i q_i^T$ should not be added. Suppose that for $i = 1, \ldots, 2m$, all dyads $s_i q_i q_i^T$ such that 3.3.1 holds are not added. Then the resulting matrix $H^{(2m)}$ is positive definite. If at least one of the dyads $s_i q_i q_i^T$ $(i = 1, \ldots, 2m)$ has been added, and then the dyads $-\eta\, e_i e_i^T$ $(i = 1, \ldots, n)$ are added, then the resulting matrix $H^{(r)}$ is a positive definite

approximation to $G^{(k)^{-1}}$ which is exact if all of the dyads $s_i q_i q_i^T$ $(i = 1, \ldots, 2m)$ are added.

If 3.3.1 holds for $i = 1, \ldots, 2m$ then $G^{(k)}$ is replaced with the $n \times n$ unit matrix I and a steepest descent step is taken.

The preceding ideas are essentially modification M1 of Sisser's method (see Figure 3.10.5). A considerable saving in computational labour can be made by omitting to add the dyads $s_i q_i q_i^T$ for which

$$|s_i| q_i^T q_i < \varepsilon_M, \qquad 3.3.2$$

where ε_M is the smallest machine number such that $1 + \varepsilon_M > 1$. This is essentially modification M2 (see Figure 3.10.6).

3.4 The Modifications M3 and M4 of Sisser's Method

In S and in the modifications S1, S2, S3, M1, and M2, the values of $s_i \in R$ and of $q_i \in R^n$ $(i = 1, \ldots, 2m)$ are all computed. Now in M2, only those dyads $s_i q_i q_i^T$ for which 3.3.2 does not hold are added. This suggests that a further saving of computational labour would result if <u>all</u> of the s_i $(i = 1, \ldots, 2m)$ were evaluated, but that only those q_i $(i = 1, \ldots, 2m)$ for which $s_i > 0$ were evaluated, and that only the dyads $s_i q_i q_i^T$ for which $s_i > 0$ were added. This constitutes modification M3 of Sisser's method (see Figure 3.10.7). If, as in M2, all small dyads are excluded from being absorbed in M3, then this constitutes modification M4 (see Figure 3.10.8).

3.5 The modifications M5 and M6 of Sisser's Method

The ideas in M1 and M3 can be combined as follows. First, absorb all dyads $s_p q_p q_p^T$,

where

$$p \in \{i \mid s_i > 0 \quad (1 \leq i \leq 2m)\} = \Phi$$

by using 3.2.11–3.2.13. By Theorem 3.2.4 the resulting matrix $H^{(p)}$ for some $p \in \Phi$ is positive definite. Then we try to absorb any dyad $s_\tau q_\tau q_\tau^T$, where

$$\tau \in \{i \mid s_i < 0 \quad (1 \leq i \leq 2m)\} = \Omega$$

which satisfies the test of positive definiteness (Theorem 3.2.4). Then the final matrix $H^{(2m)}$ is positive definite. If there is no dyads to be absorbed then the strategy used in M1 is employed. This is essentially modification M5 (see Figure 3.10.9).

If, as in M2 and M4, the dyads which are to be absorbed in M5 are those which do not satisfy 3.3.2 then modification M6 is obtained (see Figure 3.10.10).

3.6 The modifications M7, M8, M9 and M10 of Sisser's Method

The dyads given by 3.2.14 are ordered [Sis---82a] as in 3.2.8 before inversion takes place so that the positive scalars s_i ($i \in \Phi$) come first. This means that the ordering is imposed on <u>all</u> r dyads where r is as in 3.2.14. We now impose the ordering 3.2.8 on the dyads $s_\tau q_\tau q_\tau^T$ ($\tau \in \Omega$) only. The dyads $s_p q_p q_p^T$ ($p \in \Phi$) can easily be indentified and be absorbed first using 3.2.11–3.2.13, which gives a positive definite matrix $H^{(p)}$. Let k and ν be the total number of elements in Φ and Ω respectively. Then $k + \nu = r$. Now 3.2.14 can be rewritten as

$$G = D + \sum_{p \in \Phi} s_p q_p q_p^T + \sum_{i=1}^{l} s_{\tau_i} q_{\tau_i} q_{\tau_i}^T + \sum_{i=l+1}^{\nu} s_{\tau_i} q_{\tau_i} q_{\tau_i}^T$$

in which upon adding the $(l+1)^{\text{th}}$ ordered dyad the test of positive definiteness fails. Sisser's methods S, S1, S2 and S3 can easily be adapted to these changes. As in S and S1, case (i) of §3.2 might arise if $k+l < n$. The modifications M7 and M8 are then as shown in Figures 3.10.11 and 3.10.12.

Modifications M9 and M10 follow as in S2 and S3 respectively, where in M9 and M10, $s_p q_p q_p^T$ $(p \in \Phi)$ are absorbed first before $s_\tau q_\tau q_\tau^T$ $(\tau \in \Omega)$ are ordered and absorbed accordingly. The modifications M9 and M10 are shown in Figures 3.10.13 and 3.10.14.

3.7 The Modifications M11 and M12 of Sisser's Method

The dyads to be absorbed are as in M7 but they are not all re-ordered and are those which do not satisfy 3.3.2. When we add the k^{th} dyad we compute $\mu^{(k)}$ from 3.2.19 where the p_i $(i = k, \ldots, \nu)$ are determined as follows. If, by using the suggested value of δ [Sis---82a] (see §3.12) we obtain from 3.2.18 $p_i > 1$ then we take $p_i = 0.5$ and δ is computed from

$$\delta = 1 + 0.5 s_i q_i^T H^{(i-1)} q_i. \qquad 3.7.1$$

This guarantees the positive value of $\mu^{(k)}$ in 3.2.19. Furthermore the denominator of 3.2.18 is small enough to give $p_i > 1$. There is no need to test positive definiteness when absorbing dyads $p_i s_i q_i q_i^T$ because the value of the new δ in 3.7.1 is positive. The other parts of the dyads, namely $(1 - p_i) s_i q_i q_i^T$ $(i = k, \ldots, \nu)$ are not absorbed in case (ii) of §3.2. This constitutes modification M11 (see Figure 3.10.15).

Now, the dyads to be absorbed are as in M8 but they are not all re-ordered and are those which do not satisfy 3.3.2. If the test of positive definiteness fails at dyad i then we compute

$$\mu_i^{(k)} = |s_i| q_i^T q_i \quad (i \in \Omega);$$

otherwise the dyad can be absorbed. Suppose that all dyads $s_i q_i q_i^T$ $(i \in \Omega)$ have been tested and been absorbed if suitable. Then from 3.2.16 we have

$$\mu^{(k)} = \sum_{i \in \Omega} \mu_i^{(k)} \tag{3.7.2}$$

which is the sum of values $\mu_i^{(k)}$ for which the test of positive definiteness on dyad $s_i q_i q_i^T$ $(i \in \Omega)$ fails. This constitutes modification M12 (see Figure 3.10.16).

3.8 The Completing Method (C)

Let $A, B \in R^{n \times n}$ be such that A^{-1} exists, and B differs from A in column k only, so that if

$$A = \left(a_{.1} \vdots \cdots \vdots a_{.k} \vdots \cdots \vdots a_{.n}\right) \tag{3.8.1}$$

and

$$B = \left(b_{.1} \vdots \cdots \vdots b_{.k} \vdots \cdots \vdots b_{.n}\right), \tag{3.8.2}$$

where $a_{.k}$ and $b_{.k}$ are the k^{th} columns of A and B respectively. Then

$$a_{.j} = b_{.j} \quad (j = 1, \ldots, n; j \neq k), \tag{3.8.3}$$

and for some $c_{.k} \in R^n$,

$$b_{.k} = a_{.k} + c_{.k}. \tag{3.8.4}$$

Let $e_{.k}$ be column k of the $n \times n$ unit matrix. Then by 3.8.1–3.8.4,

$$B = A + c_{.k}\, e_{.k}^{T}. \tag{3.8.5}$$

Let

$$\gamma = 1 + y_k, \tag{3.8.6}$$

where y_k is the k^{th} element of

$$y = A^{-1} c_{.k}, \tag{3.8.7}$$

and let

$$z^{T} = h_{k.}, \tag{3.8.8}$$

where $h_{k.}$ is the k^{th} row of

$$A^{-1} = H$$

$$= \begin{pmatrix} h_{1.} \\ \cdots \\ \vdots \\ \cdots \\ h_{n.} \end{pmatrix}. \tag{3.8.9}$$

If $\gamma \neq 0$ then by Theorem 3.2.1, B^{-1} exists and

$$B^{-1} = H - (1/\gamma)\, yz^{T}. \tag{3.8.10}$$

The formula 3.8.10 may be used to invert a nonsingular matrix $A = (a_{ij})_{n\times n}$ as follows.

Let

$$A^{(0)} = diag(a_{11}, \ldots, a_{nn}) \qquad 3.8.11$$

and suppose that $a_{ii} \neq 0$ $(i = 1, \ldots, n)$. Then

$$H^{(0)} = A^{(0)-1}$$

$$= diag(1/a_{11}, \ldots, 1/a_{nn}). \qquad 3.8.12$$

For $k = 1, \ldots, n$ let

$$c_{.k} = a_{.k} - a_{kk}\, e_{.k}, \qquad 3.8.13$$

let

$$A^{(k)} = A^{(k-1)} + c_{.k}\, e_{.k}^{T}, \qquad 3.8.14$$

and let

$$H^{(k)} = A^{(k)-1}. \qquad 3.8.15$$

Then $A^{(n)} = A$, $H^{(n)} = A^{-1}$, and by 3.8.10, for $k = 1, \ldots, n$,

$$H^{(k)} = H^{(k-1)} - \left(1/\gamma^{(k)}\right) y^{(k)} z^{(k)T}, \qquad 3.8.16$$

where

$$H^{(k-1)} = \begin{pmatrix} h_{1.}^{(k-1)} \\ \cdots \\ \vdots \\ \cdots \\ h_{n.}^{(k-1)} \end{pmatrix}, \qquad 3.8.17$$

$$y^{(k)} = H^{(k-1)} c_{.k}, \qquad 3.8.18$$

$$z^{(k)T} = h_{k.}^{(k-1)}, \qquad 3.8.19$$

where $h_{k.}^{(k-1)}$ is the k^{th} row of $H^{(k-1)}$ and

$$\gamma^{(k)} = 1 + y_k^{(k)}. \qquad 3.8.20$$

Therefore if A is nonsingular and $a_{ii} \neq 0$ $(i = 1, \ldots, n)$ then A^{-1} may be determined by computing $H^{(k)}$ $(k = 1, \ldots, n)$ from 3.8.16, from which A^{-1} is determined as $H^{(n)}$.

The following conjecture is crucial in the use of 3.8.16 to determine a positive definite matrix $\tilde{G}^{(k)^{-1}}$ for use in 3.1.4.

Conjecture 3.8.1

Let $\Delta^{(k)}$ $(k = 1, \ldots, n)$ be the determinants of the principal submatrices of $A \in R^{n \times n}$, and let $\gamma^{(k)}$ $(k = 1, \ldots, n)$ be defined by 3.8.20. Then for $k = 1, \ldots, n$

$$\gamma^{(k)} = \frac{\Delta^{(k)}}{\Delta^{(k-1)} a_{kk}},$$

where $\Delta^{(0)} = 1$. □

Conjecture 3.8.1 is proved in Appendix B.

Now $A \in R^{n\times n}$ is positive definite if and only if for $k = 1,\ldots,n$, $\Delta^{(k)} > 0$. Furthermore if A is positive definite then for $k = 1,\ldots,n$, $a_{kk} > 0$. So if A is positive definite then by Conjecture 3.8.1, $\gamma^{(k)} > 0$ $(k = 1,\ldots,n)$. Furthermore A^{-1} is positive definite and is given by $A^{-1} = H^{(n)}$. Therefore if in 3.1.3 $G^{(k)}$ is positive definite then 3.8.16 may be used to determine the positive definite matrix $G^{(k)^{-1}}$. If however, $G^{(k)}$ is not positive definite then 3.8.16 can still be used to obtain a positive definite matrix $\tilde{G}^{(k)^{-1}}$ for use in 3.1.4 as explained subsequently.

Suppose that $a_{ii} > 0$ $(i = 1,\ldots,n)$ and that for some $k \in \{2,\ldots,n\}$, $\gamma^{(i)} > 0$ $(i = 1,\ldots,k-1)$ and $\gamma^{(k)} \leq 0$. Then by Conjecture 3.8.1, A is not positive definite, but $\Delta^{(i)} > 0$ $(i = 1,\ldots,k-1)$. This suggests that a_{kk} should be replaced with $\bar{a}_{kk}$, where

$$\bar{a}_{kk} = a_{kk} + \mu_k \qquad 3.8.21$$

in which $\mu_k > 0$ is such that $\bar{\gamma}^{(k)} > 0$, where $\bar{\gamma}^{(k)}$ is obtained by replacing a_{kk} with $\bar{a}_{kk}$.

3.9 The Product Method (P)

Let $A, B \in R^{n\times n}$ be such that A^{-1} exists, and B differs from A only in column k, so that 3.8.1–3.8.4 hold. It is required to determine B^{-1}, if it exists, knowing A^{-1}. Let

$$y = A^{-1} b_{.k}\,, \qquad 3.9.1$$

where $b_{.k}$ is column k of B and let

$$z = \left(\frac{-y_1}{y_k}, \ldots, \frac{-y_{k-1}}{y_k}, \frac{1}{y_k}, \frac{-y_{k+1}}{y_k}, \ldots, \frac{-y_n}{y_k}\right)^T. \qquad 3.9.2$$

If B^{-1} exists then $y_k \neq 0$, for if $y_k = 0$ then the columns of B are linearly dependent whence $rank(B) < n$. So if B^{-1} exists then z given by 3.9.2 is defined. Let $E \in R^{n\times n}$ be defined by

$$E = \left(e_{.1} \vdots \cdots \vdots e_{.k-1} \vdots z \vdots e_{.k+1} \vdots \cdots \vdots e_{.n}\right), \qquad 3.9.3$$

where for $j = 1, \ldots, n$, $e_{.j}$ is column j of the $n \times n$ unit matrix I. Then by 3.9.1–3.9.3

$$EA^{-1}B = CD, \qquad 3.9.4$$

where $C = \left(c_{ij}\right)_{n\times n}$ and $D = \left(d_{ij}\right)_{n\times n}$ are defined by

$$c_{ij} = \begin{cases} \delta_{ij} & (j \neq k) \\ -\frac{y_i}{y_k} & (i \neq j = k) \\ \frac{1}{y_k} & (i = j = k) \end{cases} \qquad d_{ij} = \begin{cases} \delta_{ij} & (j \neq k) \\ y_i & (i \neq j = k), \\ y_k & (i = j = k) \end{cases}$$

where δ_{ij} is the Kronecker delta, whence $CD = I$, so that by 3.9.4,

$$B^{-1} = EA^{-1}. \qquad 3.9.5$$

The preceding ideas may be used to compute B^{-1} where $B \in R^{n\times n}$ is nonsingular as

follows. For $k = 0, \ldots, n$ let $H^{(k)}$ be defined by

$$H^{(0)} = I,$$

$$H^{(k)} = E^{(k)} H^{(k-1)} \quad (k = 1, \ldots, n), \tag{3.9.6}$$

where

$$E^{(k)} = \left(e_{.1} \vdots \cdots \vdots e_{.k-1} \vdots z^{(k)} \vdots e_{.k+1} \vdots \cdots \vdots e_{.n} \right) \tag{3.9.7}$$

in which

$$z^{(k)} = \left(\frac{-y_1}{y_k}, \cdots, \frac{-y_{k-1}}{y_k}, \frac{1}{y_k}, \frac{-y_{k+1}}{y_k}, \cdots, \frac{-y_n}{y_k} \right)^T \tag{3.9.8}$$

and

$$y = H^{(k-1)} b_{.k}. \tag{3.9.9}$$

Then

$$B^{-1} = H^{(n)}. \tag{3.9.10}$$

If for some $k \in \{1, \ldots, n\}$, $y_k = 0$ then B is singular.

The computational labour required to determined $H^{(n)}$ from 3.9.6 may be reduced by using the following theorem.

Theorem 3.9.1

With the preceding notation, if $E^{(k)} = \left(E_{ij}^{(k)}\right)_{n \times n}$ $(k = 1, \ldots, n)$ and $H^{(k)} = \left(H_{ij}^{(k)}\right)_{n \times n}$

$(k = 0, \ldots, n)$ then

$$H_{ij}^{(k)} = \begin{cases} H_{ij}^{(k-1)} + z_i^{(k)} H_{kj}^{(k-1)} & (i \neq k; j = 1, \ldots, n), \\ z_k^{(k)} H_{kj}^{(k-1)} & (i = k; j = 1, \ldots, n), \\ 0 & (i = 1, \ldots, k; j = k+1, \ldots, n), \\ \delta_{ij} & (i = k+1, \ldots, n; j = k+1, \ldots, n). \end{cases} \quad 3.9.11$$

Proof

The result follows from 3.9.6 by evaluation of the $H_{ij}{}^{(k)}$. □

Furthermore, by Theorem 3.9.1 and 3.9.9, for $k = 1, \ldots, n$,

$$y_k = \sum_{j=1}^{k-1} H_{kj}^{(k-1)} b_{jk} + b_{kk}. \quad 3.9.12$$

If $y_k = 0$ then B is singular. If, however, a positive number μ_k is added to b_{kk} so as to make $y_k > 0$ then the procedure for calculating $H^{(k)}$ can continue. The addition of μ_k to b_{kk} is equivalent to replacing the original matrix B with $B + D^{(k)}$ where $D^{(k)} = diag(0, \ldots, 0, \mu_k, 0, \ldots, 0)$. The following conjecture, which is crucial in the use of 3.9.6 to determine a positive definite matrix $\tilde{G}^{(k)^{-1}}$ for use in 3.1.4 would, if proved to hold for arbitrary $n \geq 1$, establish that if $\mu_k > 0$ were sufficiently large to make $y_k > 0$ then the resulting matrix $H^{(n)}$ would be positive definite.

Conjecture 3.9.1

Let $\Delta^{(k)}$ $(k = 1, \ldots, n)$ be the determinants of the principal submatrices of $B \in R^{n \times n}$ and let $y^{(k)}$ $(k = 1, \ldots, n)$ be defined by

$$y^{(k)} = H^{(k-1)} b_{.k}.$$

Then for $k = 2, \ldots, n$,

$$y_k^{(k)} = \frac{\Delta_k}{\Delta_{k-1}}$$

and $y_1^{(1)} = \Delta_1$. □

Conjecture 3.9.1 is proved in Appendix C.

Theorem 3.9.2

If for $k = 0, \ldots, n$, $H^{(k)}$ is computed from 3.9.6, and $y_k^{(k)}$ is replaced with $y_k^{(k)} + \mu_k$ if $y_k^{(k)} \leq 0$, where $\mu_k > 0$ is such that $y_k^{(k)} + \mu_k > 0$, then $H^{(n)}$ is positive definite.

Proof

By Conjecture 3.9.1, for $m = 1, \ldots, n$,

$$\Delta_m = \prod_{k=1}^{m} y_k^{(k)}.$$

So if the μ_k are chosen so that $y_k^{(k)} > 0$ then for $k = 1, \ldots, n$, $\Delta_k > 0$, whence $H^{(n)}$ is positive definite. □

It follows from Theorem 3.9.2 that if $H^{(n)}$ is computed from 3.9.6 with the μ_k chosen so that $y_k^{(k)} + \mu_k > 0$ then

$$H^{(n)} = \left(B + diag(\mu_1, \ldots, \mu_n)\right)^{-1}$$

and $H^{(n)}$ is positive definite, and if for $k = 1, \ldots, n$, $y_k^{(k)} > 0$ then $H^{(n)} = B^{-1}$ and $H^{(n)}$ is positive definite. Therefore if, in 3.1.3, $G^{(k)}$ is positive definite then 3.9.6 may be used to compute the positive definite matrix $G^{(k)^{-1}}$, while if $G^{(k)}$ is not positive definite then 3.9.6 may still be used to compute the positive definite matrix $\tilde{G}^{(k)^{-1}}$ where

$$\tilde{G}^{(k)} = G^{(k)} + diag(\mu_1, \ldots, \mu_n)$$

in which for $i = 1, \ldots, n$, $\mu_i > 0$ is chosen so that $y_i^{(i)} > 0$.

3.10 The Summary of Sisser's Methods and the Modifications

1. Reorder the dyads as in 3.2.8.
2. Absorb these dyads one at a time.
3. If the test of positive definiteness fails on dyad i, then compute p(i) from 3.2.18 to determine μ in 3.2.19. Then absorb p(i) of the dyad using 3.2.12 where s(i) is replaced with p(i)s(i). The other part,i.e. (1-p(i)) of the dyad will be absorbed later in case (ii) of §3.2.
4. The dyads which are equivalent to μI where I is the unit matrix, will be absorbed either in case (i) or in case (ii) of §3.2.

S

Figure 3.10.1: Sisser's Method S.

1. Reorder the dyads as in 3.2.8.
2. If the test of positive definiteness on dyad i fails, then compute μ from 3.2.16.
3. Absorb dyads which are equivalent to μI where I is the unit matrix, either in case (i) or case (ii) of §3.2.
4. In case (i) the whole dyads are re-absorbed. In case (ii) the dyads starting from dyad i onward are absorbed.

S1

Figure 3.10.2: Sisser's Method S1.

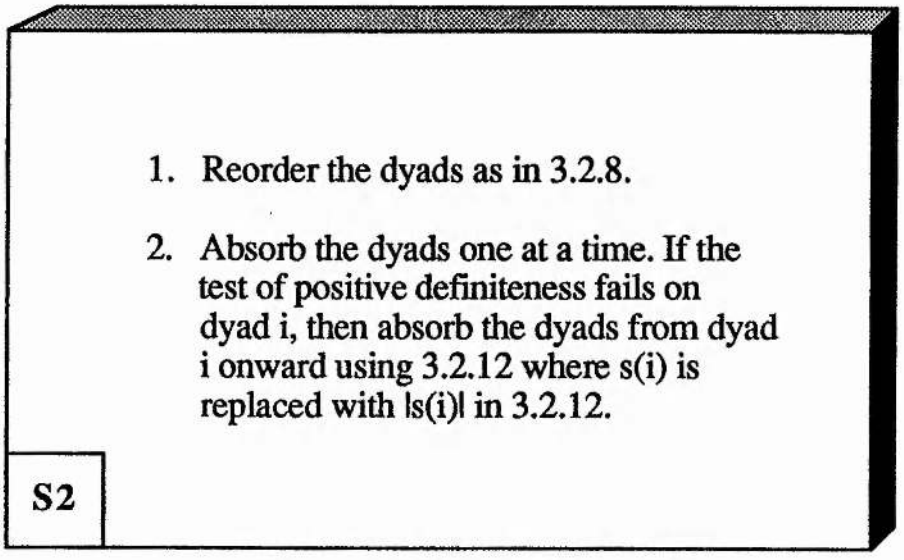


Figure 3.10.3: Sisser's Method S2.

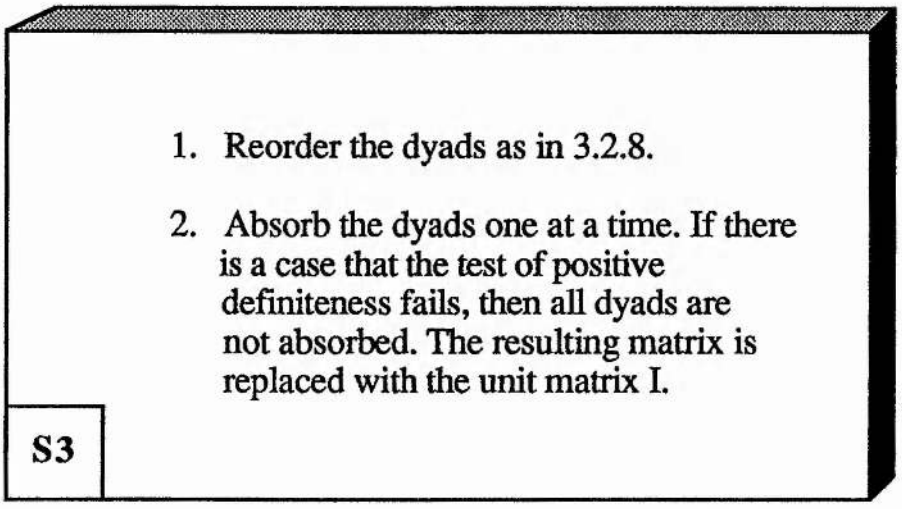


Figure 3.10.4: Sisser's Method S3.

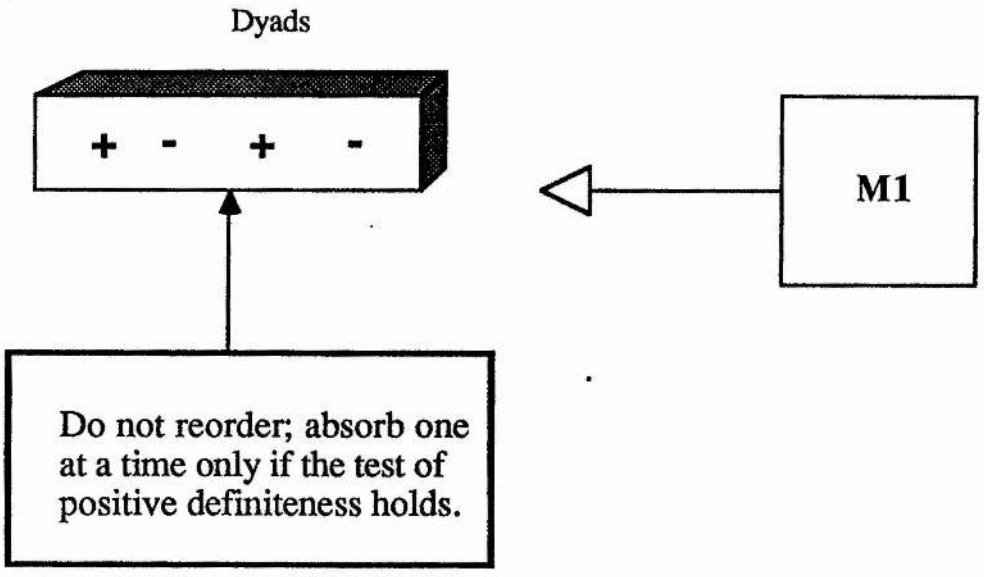


Figure 3.10.5: Modification M1.

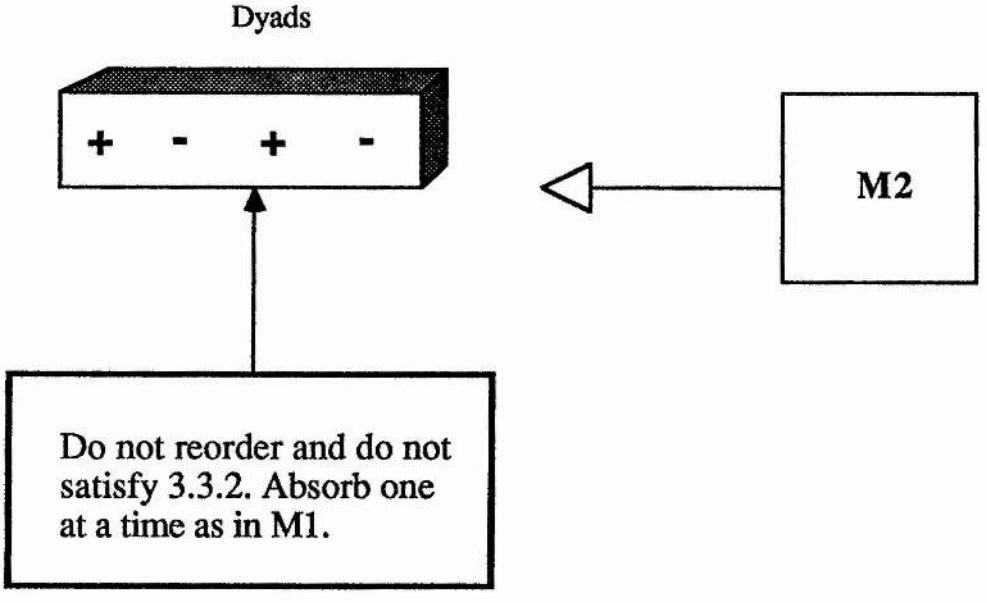


Figure 3.10.6 : Modification M2.

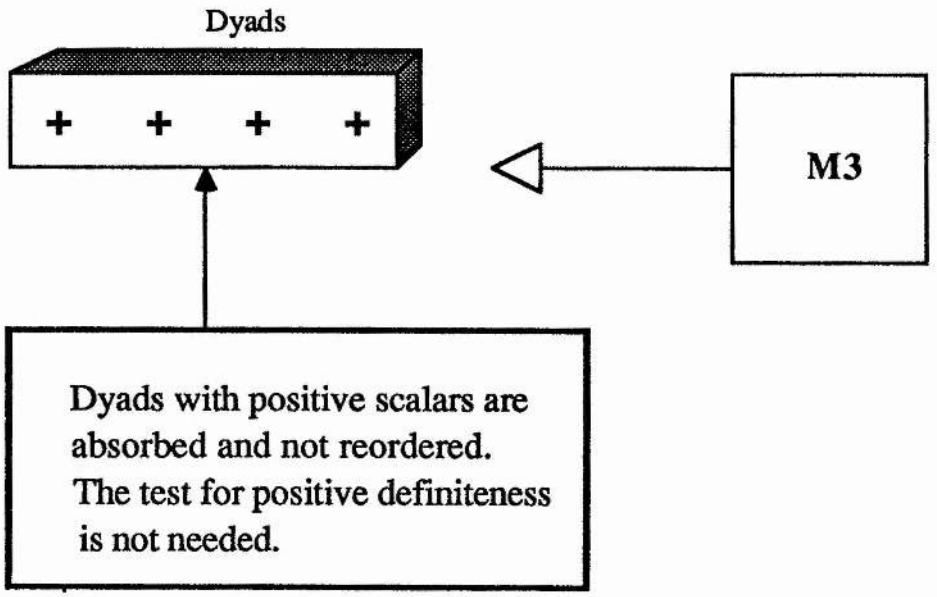


Figure 3.10.7 : Modification M3.

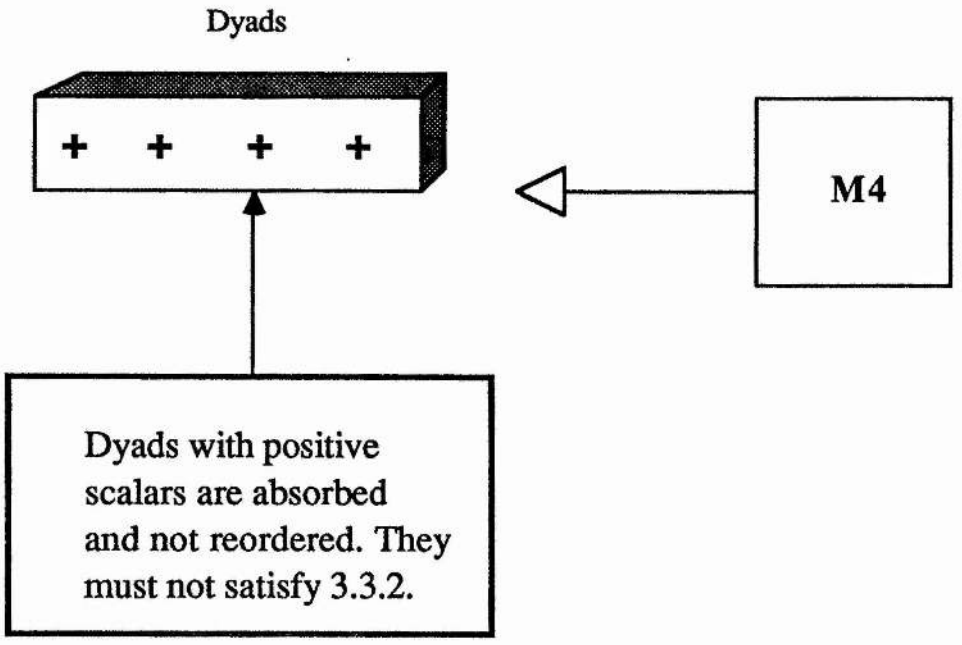


Figure 3.10.8 : Modification M4.

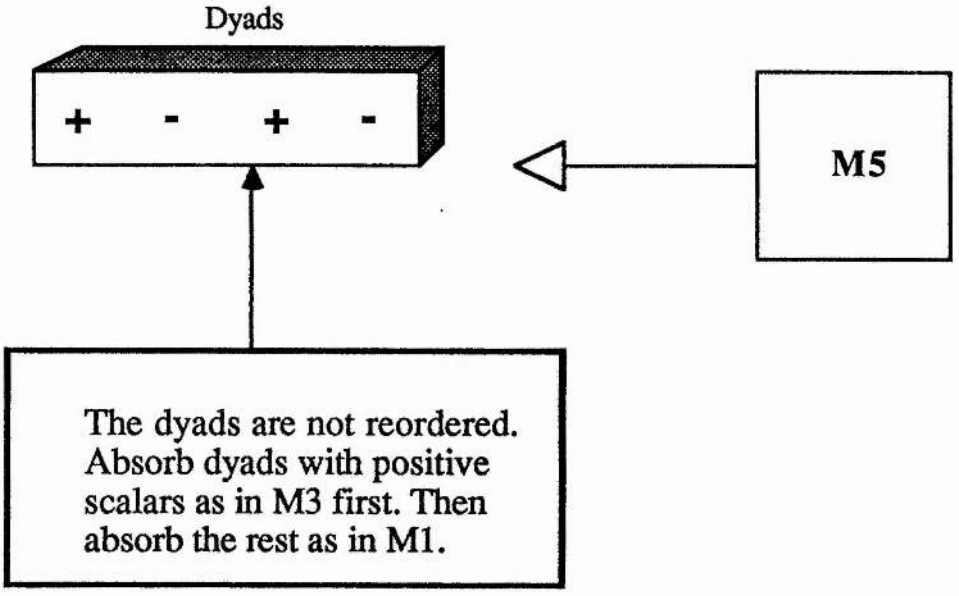


Figure 3.10.9: Modification M5.

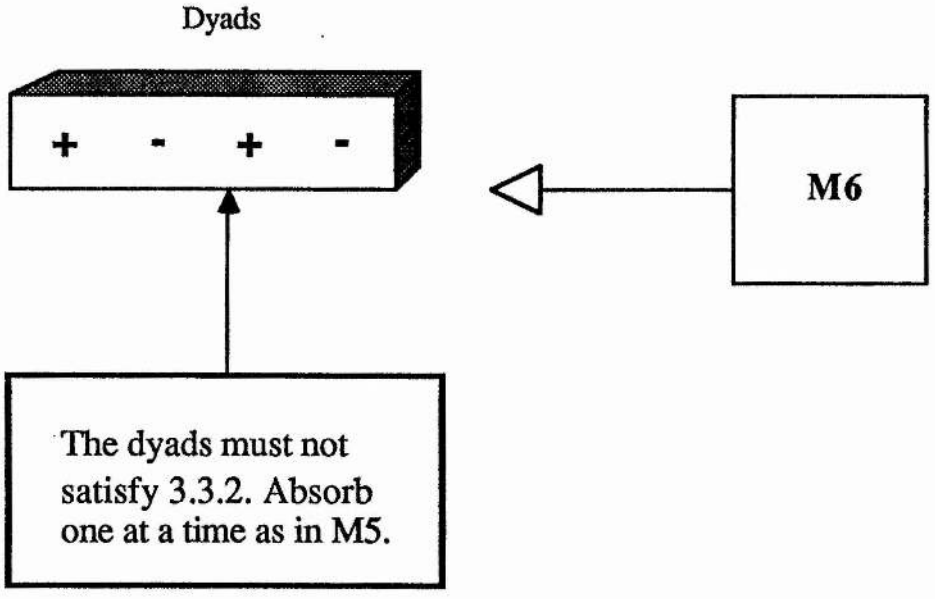


Figure 3.10.10: Modification M6.

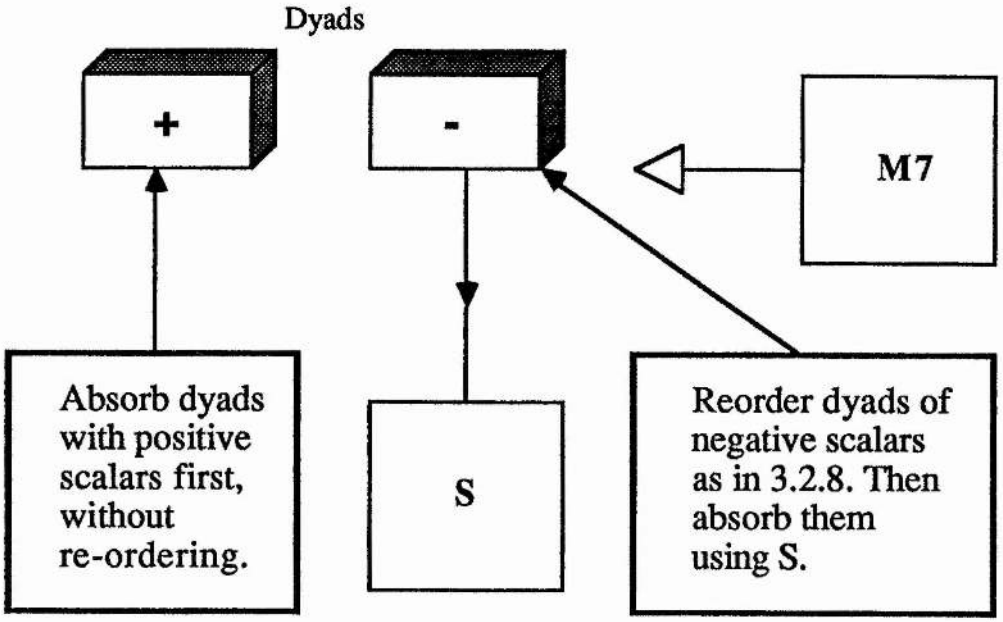


Figure 3.10.11: Modification M7.

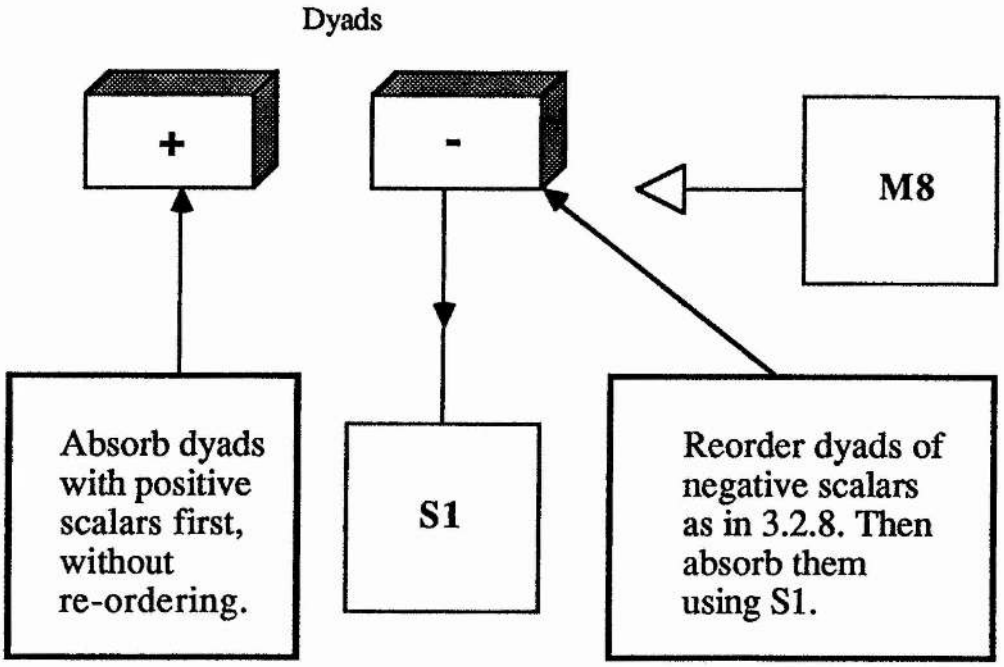


Figure 3.10.12: Modification M8.

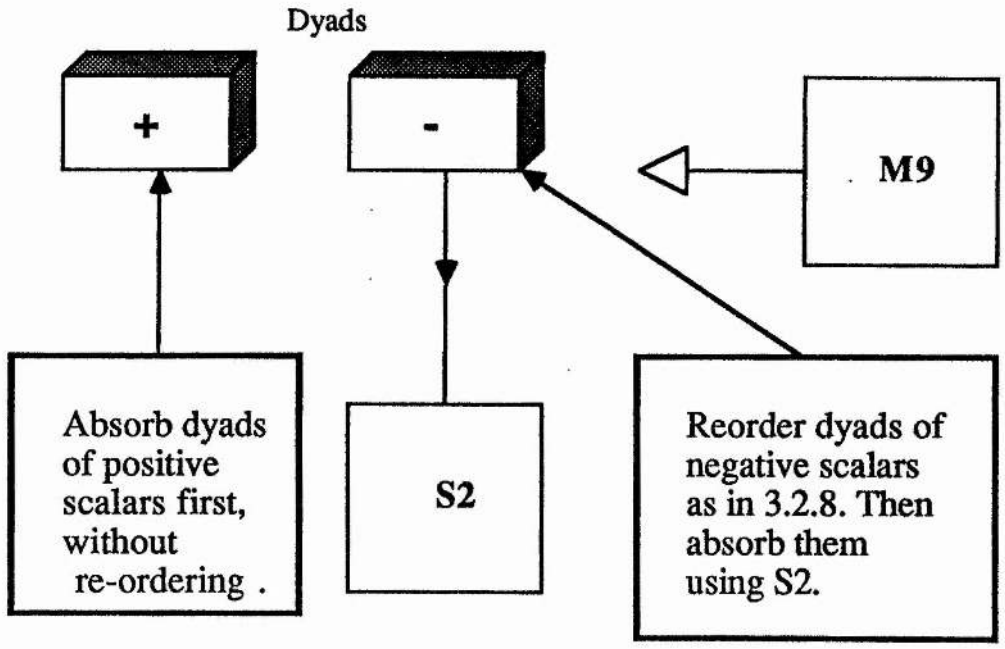


Figure 3.10.13: Modification M9.

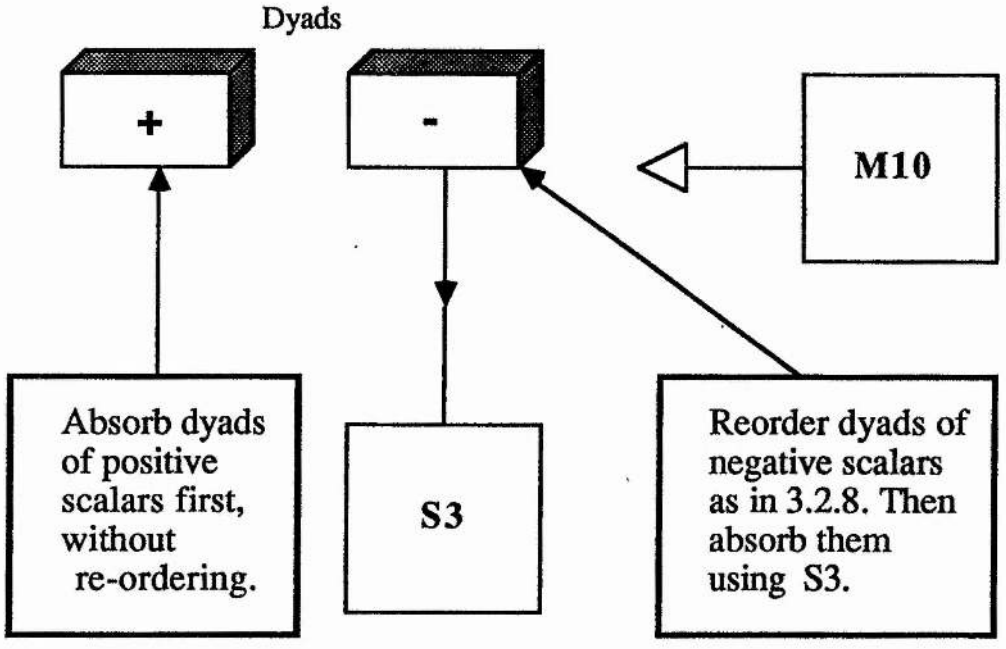


Figure 3.10.14: Modification M10.

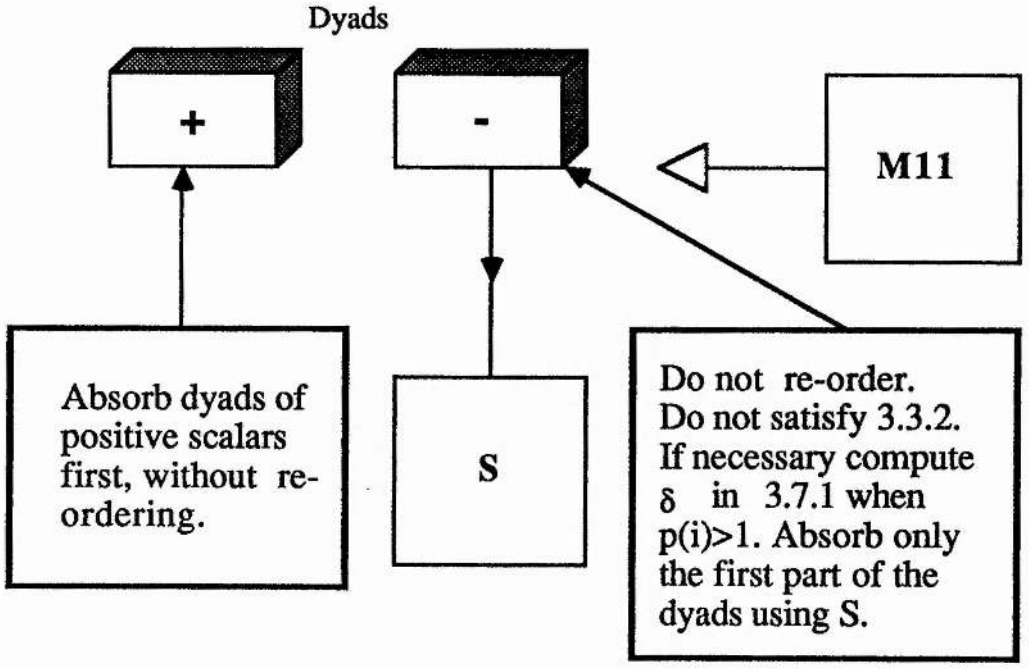


Figure 3.10.15: Modification M11.

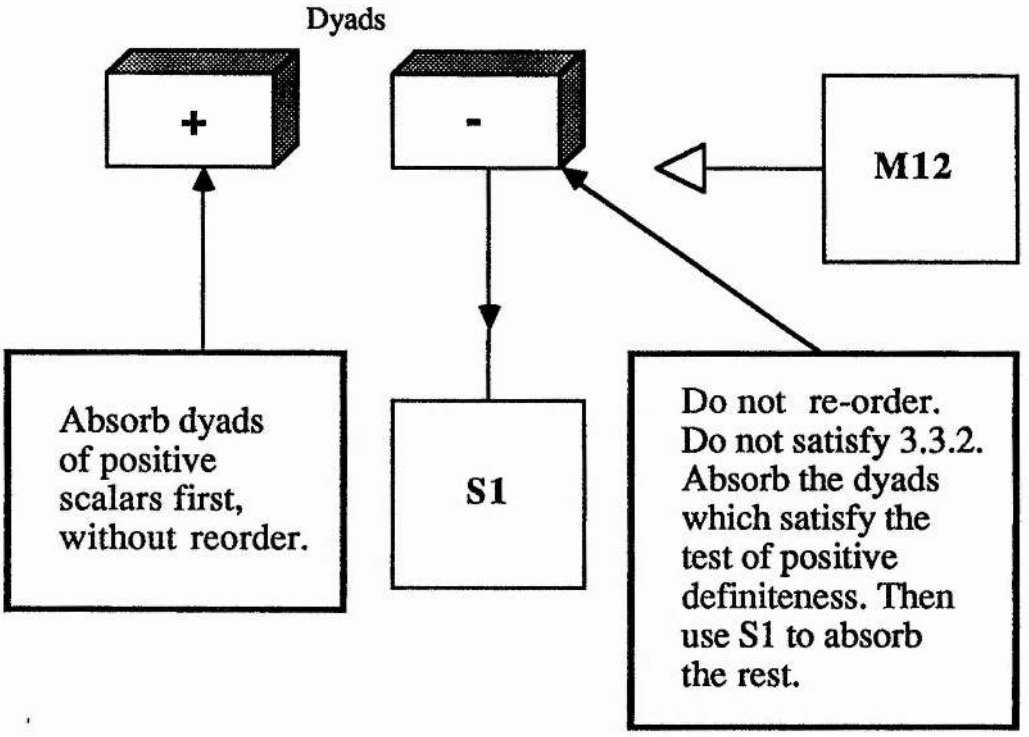


Figure 3.10.16: Modification M12.

3.11 A Flow Diagram for the Minimization Algorithms

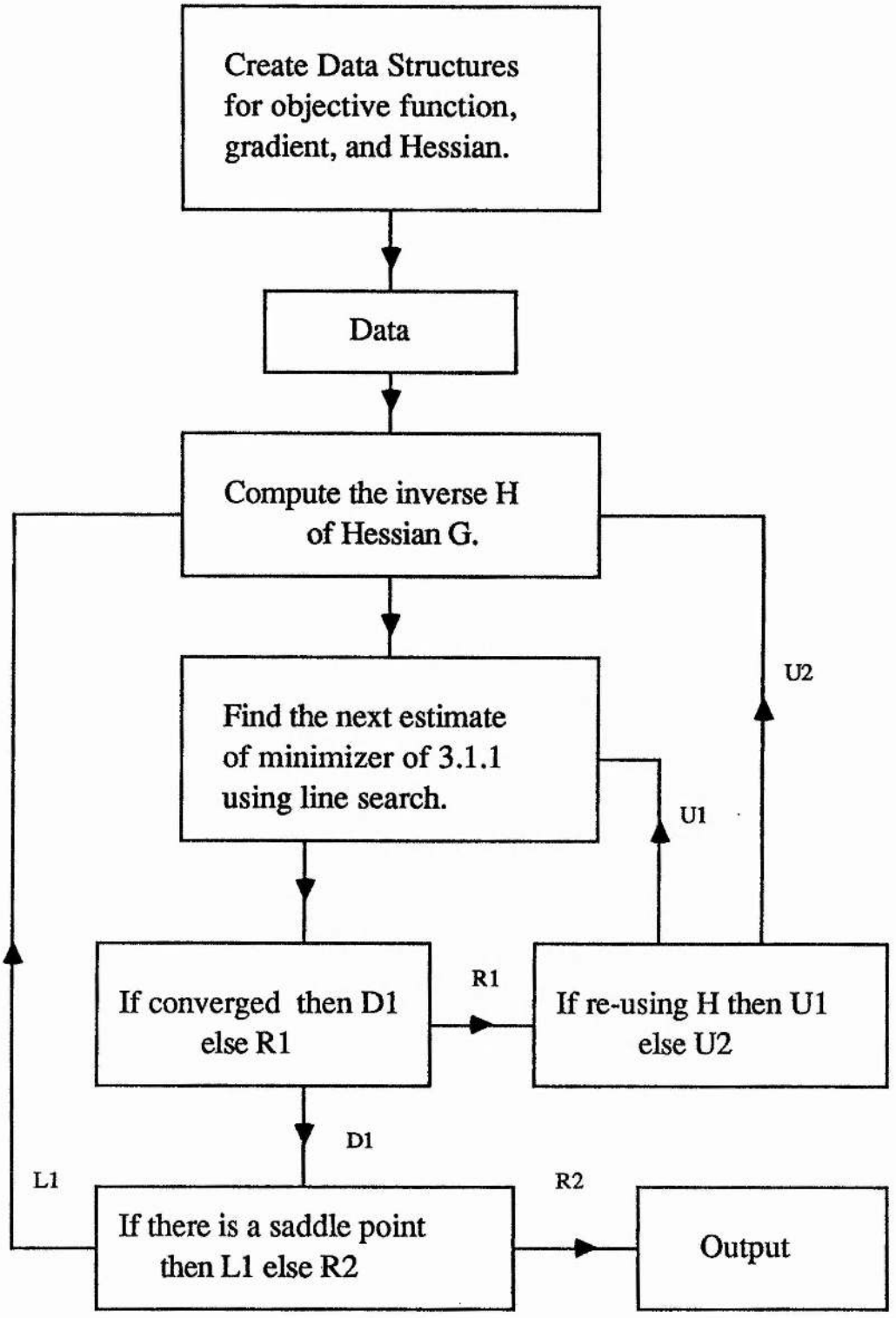


Figure 3.11.1: Flow Diagram.

3.12 Numerical Results

The procedures S, S1, S2, S3, M1–M12, C and P have been implemented in S-algol [ColM--82a] on a VAX 11-785 computer. Numerical results are presented for 7 examples (see Appendix A). The initial estimate of the minimizer for each example is given in Appendix A and the stopping criterion is

$$||g||_2 < 10^{-6},$$

where g is the gradient of the objective function at the last estimate of the minimizer. The value η used in 3.2.7 and 3.2.11 is $\eta = 0.005$. In 3.3.2, $\varepsilon_M = 10^{-16}$. As in [Sis---82a] the value δ used to construct p_i in 3.2.18 is $\delta = 0.0005$. In the method C, if $\gamma^{(k)}$ defined in 3.8.20 is non-positive then $y_k^{(k)}$ is replaced in 3.8.20 with

$$\bar{y}_k^{(k)} = r y_k^{(k)},$$

where

$$r = a_{kk}/(-a_{kk}y_k^{(k)} + \varepsilon_C),$$

in which $\varepsilon_C = 10^{-8}$. In the method P, if y_k computed from 3.9.12 is non-positive then y_k is replaced with $\bar{y}_k$ where

$$\bar{y}_k = \max\left\{|y_k|, \varepsilon_P\right\},$$

in which $\varepsilon_P = 10^{-8}$.

Tables 3.12.i.j(a) ($j = 1, 2, 3$) contain the cpu times in seconds for the methods to compute the minimizers of the test example 3.i ($i = 1, \ldots, 7$). The methods are divided into groups as shown by the vertical lines in those tables; this in view of the fact that the modifications are made from the methods in the far left of each column, save the method P. For

example, M12 and M8 are modified from the method S1. The figures in those tables are obtained from various values of m where m is the number of times the inverse Hessian is re-used. For example if $m = 2$ then the last computed inverse Hessian is re-used twice for determining the next two estimations of the minimizer. Thus, if $m = 0$ then the new inverse Hessian is computed for every iteration (see the flow diagram in §3.11). The corresponding number of iterations nI and number of function evaluations nf are recorded in Tables 3.12.i.j(b) ($j = 1, 2, 3$). The expression $a : b$ means $nI = a$ and $nf = b$.

The effectiveness of using various values of m can be observed in Table 3.12.8. The figures in the tables are obtained by counting the number of methods which attain the smallest cpu time for the four values of m ($m = 0, 1, 2, 3$). For example, 17 methods and 1 method attain the smallest cpu time for $m = 3$ and $m = 2$ respectively for Example 3.1 (see Tables 3.12.1.1(a)–3.12.1.3(a)), while for $m = 0$ and $m = 1$, no method attains the smallest cpu time.

Tables 3.12.9–3.12.15 contain the positions of the methods with respect to the cpu times in seconds for each value of m. The figures in these tables are obtained as follows. From Tables 3.12.1.1(a)–3.12.1.3(a) for Example 3.1 with $m = 0$, the smallest cpu time is attained by M3, so we mark it with number 1 in Table 3.12.9. For the second best is M4 which we mark with number 2. So the worst method for the particular test example is in the position 16, *i.e.* the method M5, since we have 16 methods of dyad form. These positions might be different with respect to m, so the figures in the last column of the tables indicate the position of each method more reliably. It shows that the methods M3, M4, M6 and M5 are in positions 1, 2, 3 and 16 respectively. Other methods will be judged accordingly. Finally, Table 3.12.16 shows the positions of the methods taking all test examples into account.

From Tables 3.12.i.j(a) ($i = 1, \ldots, 7$; $j = 1, 2, 3$), the following conclusions are drawn. In the subsequent conclusions A improves B means that the cpu time taken by the method A

is less than for method B in the same group.

(a) M11 and M7 improve S, and M11 improves M7;

(b) M12 and M8 improve S1, and M12 improves M8;

(c) M9 improves S2;

(d) M10 improves S3;

(e) M2 improves M1;

(f) M4 does not improve M3;

(g) M6 improves M5;

(h) P is superior to C;

(i) Methods P and C are more efficient than methods S, S1–S3, and M1–M2.

Table 3.12.8 suggests that it is worthwhile to re-use the inverse of the Hessian two or three times if $n > 4$, but to use it once for each iteration if $n \leq 4$.

The relations between the methods C, P, S, S1–S3 and M1–M12 from different groups can be inferred from Tables 3.12.9–3.12.6. Suppose that A < B means that the cpu time for the method A is less than that for method B, so that A is superior to B. Then the figures in the last column of Table 3.12.6 reveal the following relationships:

$$\begin{aligned} \mathrm{P} < \mathrm{C} < \mathrm{M6} < \mathrm{M3} < \mathrm{M2} < \mathrm{M11} < \mathrm{M4} < \mathrm{M12} < \mathrm{M9} < \mathrm{M10} < \\ < \mathrm{M8} < \mathrm{M7} < \mathrm{M1} < \mathrm{M5} < \mathrm{S3} < \mathrm{S} < \mathrm{S1} < \mathrm{S2} \end{aligned} \qquad 3.12.1$$

The inequalities 3.12.1 confirm the conclusions (a)–(i).

Method m	S	M7	M11	S1	M8	M12
0	461.92	433.25	423.04	462.29	432.53	423.83
1	335.91	315.45	310.32	337.01	317.77	313.21
2	296.40	280.32	275.49	297.95	279.82	275.31
3	281.57	266.25	261.19	282.75	265.51	260.21

Table 3.12.1.1(a): Example 3.1 (cpu times in sec.)

Method m	S2	M9	S3	M10	M1	M2
0	465.34	430.60	468.12	430.98	469.20	388.30
1	338.25	319.31	348.42	319.27	342.11	284.70
2	297.87	280.82	298.06	278.22	301.08	254.05
3	282.78	264.66	282.27	264.66	285.59	241.22

Table 3.12.1.2(a): Example 3.1 (cpu times in sec.)

Method m	M3	M4	M5	M6	P	C
0	269.55	274.17	469.21	388.06	137.96	163.86
1	202.27	206.40	343.50	286.81	112.49	130.25
2	183.36	187.00	301.58	252.77	105.45	120.90
3	177.72	179.96	284.84	241.05	107.28	120.81

Table 3.12.1.3(a): Example 3.1 (cpu times in sec.)

Method m	S	M7	M11	S1	M8	M12
0	20:21	20:21	20:21	20:21	20:21	20:21
1	14:29	14:29	14:29	14:29	14:29	14:29
2	12:35	12:35	12:35	12:35	12:35	12:35
3	11:42	11:42	11:42	11:42	11:42	11:42

Table 3.12.1.1(b): Number of iterations and of function evaluations for Example 3.1.

Method m	S2	M9	S3	M10	M1	M2
0	20:21	20:21	20:21	20:21	20:21	20:21
1	14:29	14:29	14:29	14:29	14:29	14:29
2	12:35	12:35	12:35	12:35	12:35	12:35
3	11:42	11:42	11:42	11:42	11:42	11:42

Table 3.12.1.2(b): Number of iterations and of function evaluations for Example 3.1.

Method m	M3	M4	M5	M6	P	C
0	20:21	20:21	20:21	20:21	20:21	20:21
1	14:29	14:29	14:29	14:29	14:29	14:29
2	12:35	12:35	12:35	12:35	12:35	12:35
3	11:42	11:42	11:42	11:42	11:42	11:42

Table 3.12.1.3(b): Number of iterations and of function evaluations for Example 3.1.

Method m	S	M7	M11	S1	M8	M12
0	94.46	85.87	83.70	94.39	86.19	83.76
1	83.25	77.72	74.77	82.88	76.88	75.73
2	81.22	75.69	74.63	81.14	75.42	74.46
3	72.00	68.63	69.98	81.46	76.98	75.72

Table 3.12.2.1(a): Example 3.2 (cpu times in sec.)

Method m	S2	M9	S3	M10	M1	M2
0	93.95	85.25	94.21	85.91	98.66	79.84
1	86.57	80.54	78.02	74.71	80.63	67.83
2	81.67	75.28	80.58	75.12	75.57	64.67
3	88.57	84.30	86.89	uf	78.08	67.70

Table 3.12.2.2(a): Example 3.2 (cpu times in sec.)
(uf = underflow).

Method m	M3	M4	M5	M6	P	C
0	64.92	66.02	94.58	77.20	32.15	36.42
1	57.59	58.18	86.20	72.96	41.14	of
2	55.97	56.68	81.47	69.94	39.06	42.11
3	59.47	59.65	85.06	74.87	42.18	48.42

Table 3.12.2.3(a): Example 3.2 (cpu times in sec.)
(of = overflow).

Method m	S	M7	M11	S1	M8	M12
0	24:33	24:33	24:33	24:33	24:33	24:33
1	18:55	18:55	18:52	18:54	18:54	18:54
2	16:68	16:68	16:68	16:68	16:68	16:68
3	13:66	13:66	13:79	14:83	14:83	14:83

Table 3.12.2.1(b): Number of iterations and of function evaluations for Example 3.2.

Method m	S2	M9	S3	M10	M1	M2
0	24:33	24:33	24:33	24:33	26:37	26:37
1	19:58	19:58	16:76	16:76	18:58	18:58
2	16:68	16:68	16:68	16:68	15:68	15:68
3	15:100	15:100	14:112	uf	14:82	14:82

Table 3.12.2.2(b): Number of iterations and of function evaluations for Example 3.2. (uf = underflow).

Method m	M3	M4	M5	M6	P	C
0	26:37	26:37	26:33	24:33	24:33	24:33
1	19:54	19:54	19:57	19:57	19:74	of
2	15:68	15:68	16:68	16:68	16:68	16:68
3	14:82	14:82	14:99	14:99	14:80	13:103

Table 3.12.2.3(b): Number of iterations and of functions evaluations for Example 3.2. (of = overflow).

Method m	S	M7	M11	S1	M8	M12
0	13.97	12.31	11.81	13.77	12.71	12.18
1	12.15	11.29	10.88	12.21	11.31	10.67
2	12.12	12.23	11.53	12.98	11.87	11.43
3	12.57	12.14	11.71	12.52	12.17	12.03

Table 3.12.3.1(a): Example 3.3 (cpu times in sec.)

Method m	S2	M9	S3	M10	M1	M2
0	14.06	12.45	13.58	12.42	12.23	11.04
1	12.60	11.41	11.96	11.38	11.19	10.43
2	12.69	11.62	12.38	11.40	11.64	10.97
3	12.75	12.20	12.63	12.21	11.80	11.69

Table 3.12.3.2(a): Example 3.3 (cpu times in sec.)

Method m	M3	M4	M5	M6	P	C
0	9.19	9.52	12.50	11.22	6.56	6.81
1	10.02	9.91	11.42	10.61	7.32	7.20
2	10.18	9.91	11.63	11.52	8.87	8.49
3	11.05	11.21	12.25	11.65	8.93	9.20

Table 3.12.3.3(a): Example 3.3 (cpu times in sec.)

Method m	S	M7	M11	S1	M8	M12
0	17:18	17:18	17:18	17:18	17:18	17:18
1	12:24	12:24	12:24	12:24	12:24	12:24
2	10:30	10:30	10:30	10:30	10:30	10:30
3	9:34	9:34	9:34	9:34	9:34	9:34

Table 3.12.3.1(b): Number of iterations and of function evaluations for Example 3.3.

Method m	S2	M9	S3	M10	M1	M2
0	17:18	17:18	17:18	17:18	17:18	17:18
1	12:24	12:24	12:24	12:24	12:24	12:24
2	10:30	10:30	10:30	10:30	10:30	10:30
3	9:34	9:34	9:34	9:34	9:34	9:34

Table 3.12.3.2(b): Number of iterations and of function evaluations for Example 3.3.

Method m	M3	M4	M5	M6	P	C
0	17:18	17:18	17:18	17:18	17:18	17:18
1	13:26	13:26	12:24	12:24	12:24	12:24
2	10:31	10:31	10:31	10:31	10:31	10:31
3	9:36	9:36	9:34	9:34	9:34	9:34

Table 3.12.3.3(b): Number of iterations and of function evaluations for Example 3.3.

Method m	S	M7	M11	S1	M8	M12
0	6.00	5.89	5.62	6.30	5.85	5.67
1	8.19	7.80	6.48	7.04	6.44	6.74
2	7.64	7.51	7.67	7.89	7.47	7.98
3	10.09	10.23	9.82	9.32	9.02	8.92

Table 3.12.4.1(a): Example 3.4 (cpu times in sec.)

Method m	S2	M9	S3	M10	M1	M2
0	6.24	5.81	6.19	5.92	5.93	5.94
1	7.54	7.46	7.66	7.63	6.47	6.44
2	7.89	7.67	8.11	7.83	7.24	7.31
3	9.82	9.52	10.27	10.28	8.47	8.30

Table 3.12.4.2(a): Example 3.4 (cpu times in sec.)

Method m	M3	M4	M5	M6	P	C
0	5.30	5.42	5.62	5.34	3.80	3.76
1	6.08	6.18	6.81	6.58	6.42	6.24
2	7.00	6.66	7.68	7.44	6.22	6.16
3	8.12	8.14	9.53	9.29	6.76	8.13

Table 3.12.4.3(a): Example 3.4 (cpu times in sec.)

Method m	S	M7	M11	S1	M8	M12
0	24:33	24:33	24:33	24:33	24:33	24:33
1	19:72	19:72	18:50	18:53	18:53	18:53
2	16:68	16:68	16:68	16:68	16:68	16:68
3	15:105	14:103	14:98	14:89	14:89	14:89

Table 3.12.4.1(b): Number of iterations and of function evaluations for Example 3.4.

Method m	S2	M9	S3	M10	M1	M2
0	24:33	24:33	24:33	24:33	24:37	26:37
1	19:58	19:58	16:76	16:76	18:58	18:58
2	16:68	16:68	16:68	16:68	15:68	15:68
3	15:100	15:100	14:112	14:112	14:82	14:82

Table 3.12.4.2(b): Number of iteration and of function evaluations for Example 3.4.

Method m	M3	M4	M5	M6	P	C
0	26:37	26:37	24:33	24:33	24:33	24:33
1	19:54	19:54	19:57	19:57	19:74	19:74
2	15:68	15:68	16:68	16:68	16:68	16:68
3	14:82	14:82	14:99	14:99	14:80	13:103

Table 3.12.4.3(b): Number of iterations and of function evaluations for Example 3.4.

Method m	S	M7	M11	S1	M8	M12
0	6.32	6.01	5.94	6.49	6.09	6.02
1	6.51	6.28	6.15	6.63	5.96	6.00
2	7.13	6.92	6.85	7.04	6.96	6.78
3	7.55	7.34	7.15	7.54	7.22	7.05

Table 3.12.5.1(a): Example 3.5 (cpu times in sec.)

Method m	S2	M9	S3	M10	M1	M2
0	6.53	6.06	6.30	6.08	11.20	11.58
1	6.48	6.13	6.41	6.26	9.06	9.07
2	6.75	6.88	6.67	6.75	8.65	8.68
3	7.36	7.16	7.45	7.13	9.84	9.83

Table 3.12.5.2(a): Example 3.5 (cpu times in sec.)

Method m	M3	M4	M5	M6	P	C
0	14.64	15.33	5.72	5.69	3.44	3.42
1	13.62	13.73	6.00	5.91	4.27	4.20
2	14.14	14.22	6.15	6.61	5.10	5.00
3	13.59	13.69	6.99	6.90	5.64	5.69

Table 3.12.5.3(a): Example 3.5 (cpu times in sec.)

Method m	S	M7	M11	S1	M8	M12
0	27:28	27:28	27:28	27:28	27:28	27:28
1	20:40	20:40	20:40	20:40	20:40	20:40
2	17:50	17:50	17:50	17:50	17:50	17:50
3	15:58	15:58	15:58	15:58	15:58	15:58

Table 3.12.5.1(b): Number of iterations and of function evaluations for Example 3.5.

Method m	S2	M9	S3	M10	M1	M2
0	27:28	27:28	27:28	27:28	44:72	44:72
1	20:40	20:40	20:40	20:40	24:72	24:72
2	17:50	17:50	17:50	17:50	19:73	19:73
3	15:58	15:58	15:58	15:58	17:88	17:88

Table 3.12.5.2(b): Number of iterations and of function evaluations for EXample 3.5.

Method m	M3	M4	M5	M6	P	C
0	63:107	63:107	27:28	27:28	27:28	27:28
1	38:119	63:119	20:40	20:40	20:40	20:40
2	29:131	29:131	17:50	17:50	17:50	17:50
3	23:130	23:130	15:58	15:58	15:58	15:58

Table 3.12.5.3(b): Number of iterations and of function evaluations for Example 3.5.

Method m	S	M7	M11	S1	M8	M12
0	2.28	2.15	1.78	2.31	2.30	2.50
1	1.62	1.55	1.61	1.65	1.60	1.56
2	1.51	1.51	1.59	1.49	1.46	1.39
3	1.75	1.67	1.75	1.77	1.72	1.61

Table 3.12.6.1(a): Example 3.6 (cpu times in sec.)

Method m	S2	M9	S3	M10	M1	M2
0	2.05	2.09	1.87	1.75	1.49	1.43
1	1.63	1.56	1.78	1.77	1.71	1.72
2	1.49	1.15	1.99	1.90	1.86	1.86
3	1.73	1.70	2.10	2.04	2.08	2.05

Table 3.12.6.2(a): Example 3.6 (cpu times in sec.)

Method m	M3	M4	M5	M6	P	C
0	1.28	1.26	1.77	1.76	1.83	2.07
1	1.48	1.50	2.07	2.00	2.89	3.87
2	1.87	1.89	2.42	2.36	3.66	5.00
3	2.07	2.06	2.53	2.64	4.40	7.26

Table 3.12.6.3(a): Example 3.6 (cpu times in sec.)

Method m	S	M7	M11	S1	M8	M12
0	7:10	7:10	6:8	6:13	6:13	8:13
1	4:8	4:8	4:9	4:8	4:8	4:8
2	3:9	3:9	3:10	3:9	3:9	3:9
3	3:11	3:11	3:11	3:11	3:11	3:11

Table 3.12.6.1(b): Number of iterations and of function evaluations for Example 3.6.

Method m	S2	M9	S3	M10	M1	M2
0	6:10	7:11	6:8	6:8	5:7	5:7
1	4:8	4:8	4:11	4:11	4:11	4:11
2	3:9	3:9	3:15	3:15	3:14	3:14
3	3:11	3:11	3:17	3:17	3:17	3:17

Table 3.12.6.2(b): Number of iterations and of function evaluations for Example 3.6.

Method m	M3	M4	M5	M6	P	C
0	5:7	5:7	5:11	5:11	6:15	5:21
1	4:10	4:10	4:15	4:15	5:28	5:42
2	4:14	4:14	4:19	4:19	4:41	4:63
3	3:17	3:17	3:22	3:22	4:49	4:90

Table 3.12.6.3(b): Number of iterations and of function evaluations for Example 3.6.

Method m	S	M7	M11	S1	M8	M12
0	7.34	6.69	5.13	7.24	6.51	5.86
1	6.91	6.36	6.32	6.29	5.86	5.64
2	6.85	6.37	6.44	6.79	6.49	6.38
3	7.12	6.49	6.69	7.34	6.85	6.70

Table 3.12.7.1(a): Example 3.7 (cpu times in sec.)

Method m	S2	M9	S3	M10	M1	M2
0	7.17	6.29	5.41	4.97	5.55	5.63
1	6.38	5.82	4.62	4.19	5.50	5.51
2	6.47	6.03	4.60	4.09	6.63	6.06
3	7.26	6.43	4.90	4.73	6.51	6.66

Table 3.12.7.2(a): Example 3.7 (cpu times in sec.)

Method m	M3	M4	M5	M6	P	C
0	7.55	7.46	5.64	5.62	3.25	4.31
1	6.77	6.97	5.40	5.21	4.38	5.49
2	7.72	7.74	6.16	6.52	5.32	7.26
3	8.18	8.04	6.69	6.73	6.14	8.68

Table 3.12.7.3(a): Example 3.7 (cpu times in sec.)

Method m	S	M7	M11	S1	M8	M12
0	15:16	15:16	13:14	15:16	15:16	14:15
1	11:22	11:22	11:22	10:21	10:21	10:21
2	9:26	9:26	9:27	9:26	9:26	9:28
3	9:30	8:30	8:31	8:30	8:30	8:32

Table 3.12.7.1(b): Number of iterations and of function evaluations for Example 3.7.

Method m	S2	M9	S3	M10	M1	M2
0	15:17	15:17	11:13	11:13	14:15	14:15
1	10:20	10:20	7:16	7:16	10:20	10:20
2	8:25	8:25	6:18	6:18	9:29	9:29
3	8:30	8:30	5:22	5:22	8:31	8:31

Table 3.12.7.2(b): Number of iterations and of function evaluations for Example 3.7.

Method m	M3	M4	M5	M6	P	C
0	23:24	23:24	14:15	14:15	15:18	15:27
1	15:31	15:31	10:20	10:20	11:27	11:43
2	12:37	12:37	9:29	9:29	9:35	9:59
3	10:43	10:43	8:31	8:31	8:44	8:73

Table 3.12.7.3(b): Number of iterations and of function evaluations for Example 3.7.

Example	m n	0	1	2	3
3.1	20	0	0	1	17
3.2	10	2	2	11	3
3.3	4	4	13	1	0
3.4	2	18	0	0	0
3.5	2	11	3	2	2
3.6	2	9	1	8	0
3.7	2	3	12	3	0

Table 3.12.8: The effectiveness of re-using the inverse of the Hessian.

m Method	0	1	2	3	Total T(1)
S	13	13	13	13	52
S1	14	14	15	15	58
S2	15	15	14	16	60
S3	16	18	16	14	64
M1	17	16	17	18	68
M2	6	5	6	6	23
M3	3	3	3	3	12
M4	4	4	4	4	16
M5	18	17	18	17	70
M6	5	6	5	5	21
M7	12	9	11	12	44
M8	11	10	10	11	42
M9	9	12	12	9	42
M10	10	11	9	10	40
M11	7	7	8	8	30
M12	8	8	7	7	30
C	2	2	2	2	8
P	1	1	1	1	4

Table 3.12.9: Positions of the methods with respect to the cpu time in sec. for Example 3.1.

m Method	0	1	2	3	Total T(2)
S	16	15	16	8	55
S1	15	14	15	13	57
S2	13	18	18	17	66
S3	14	11	14	16	55
M1	18	13	12	12	55
M2	6	4	5	5	20
M3	3	2	3	3	11
M4	4	3	4	4	15
M5	17	16	17	15	65
M6	5	5	6	9	25
M7	10	10	13	6	39
M8	12	9	11	11	43
M9	9	12	10	14	45
M10	11	6	9	18	44
M11	7	7	8	7	29
M12	8	8	7	10	33
C	2	17	2	2	23
P	1	1	1	1	4

Table 3.12.10: Positions of the methods with respect to the cpu time in sec. for Example 3.2.

m Method	0	1	2	3	Total T(3)
S	17	16	14	16	63
S1	16	17	18	15	66
S2	18	18	17	18	71
S3	15	15	16	17	63
M1	9	9	12	8	38
M2	5	5	5	6	21
M3	3	4	4	3	14
M4	4	3	3	4	14
M5	13	14	11	14	52
M6	6	6	8	5	25
M7	10	10	15	10	45
M8	14	11	13	11	49
M9	12	13	10	12	47
M10	11	12	6	13	42
M11	7	8	9	7	31
M12	8	7	7	9	31
C	2	1	1	2	6
P	1	2	2	1	6

Table 3.12.11: Positions of the methods with respect to the cpu time in sec. for Example 3.3.

m Method	0	1	2	3	Total T(4)
S	15	18	10	15	58
S1	18	12	14	10	54
S2	17	14	16	14	61
S3	16	16	18	17	67
M1	14	7	5	6	32
M2	13	5	6	5	29
M3	3	1	4	2	10
M4	5	2	3	4	14
M5	6	11	13	12	42
M6	4	9	7	9	29
M7	11	17	9	16	53
M8	10	6	8	8	32
M9	9	13	12	11	45
M10	12	15	15	18	60
M11	7	8	11	13	39
M12	8	10	17	7	42
C	1	3	1	3	8
P	2	4	2	1	9

Table 3.12.12: Positions of the methods with respect to the cpu time in sec. for Example 3.4.

m Method	0	1	2	3	Total T(5)
S	12	13	14	14	53
S1	13	14	13	13	53
S2	14	12	7	11	44
S3	11	11	5	12	39
M1	15	15	15	16	61
M2	16	16	16	15	63
M3	17	17	17	17	68
M4	18	18	18	18	72
M5	4	5	3	4	16
M6	3	3	4	3	13
M7	6	10	11	10	37
M8	10	4	12	9	35
M9	8	7	10	8	29
M10	9	9	6	6	30
M11	5	8	9	7	29
M12	7	6	8	5	26
C	1	1	1	2	5
P	2	2	2	1	7

Table 3.12.13: Positions of the methods with respect to the cpu time in sec. for Example 3.5.

m Method	0	1	2	3	Total T(6)
S	15	8	7	7	37
S1	17	10	4	8	39
S2	11	9	5	5	30
S3	10	14	14	14	52
M1	4	11	9	13	37
M2	3	12	10	10	35
M3	2	2	11	12	27
M4	1	1	12	11	25
M5	7	16	16	15	54
M6	6	15	15	16	52
M7	14	3	6	2	25
M8	16	6	3	4	29
M9	13	4	1	3	21
M10	5	13	13	9	40
M11	8	7	8	6	29
M12	18	5	2	1	26
C	12	18	18	18	66
P	9	17	17	17	60

Table 3.12.14: Positions of the methods with respect to the cpu time in sec. for Example 3.6.

m Method	0	1	2	3	Total T(7)
S	16	17	15	13	61
S1	15	12	14	15	56
S2	14	15	10	14	53
S3	5	3	2	2	12
M1	6	7	13	6	32
M2	8	8	5	7	28
M3	18	16	17	17	68
M4	17	18	18	16	69
M5	9	6	6	8	29
M6	7	4	12	11	34
M7	13	14	7	5	39
M8	12	11	11	12	46
M9	11	10	4	4	29
M10	3	1	1	1	6
M11	4	13	9	9	35
M12	10	9	8	10	37
C	2	5	16	18	41
P	1	2	3	3	9

Table 3.12.15: Positions of the methods with respect to the cpu time in sec. for Example 3.7.

Method	$\sum_{j=1}^{7} T(j)$	Positions of the methods
S	379	16
S1	383	17
S2	385	18
S3	352	15
M1	323	13
M2	219	5
M3	210	4
M4	225	8
M5	328	14
M6	199	3
M7	282	12
M8	276	11
M9	258	9
M10	262	10
M11	222	6
M12	225	7
C	157	2
P	99	1

Table 3.12.16: Positions of the methods with respect to cpu time in all examples.

3.13 Future Work

Sisser [Sis---82b][Sis---82c] has described a technique for inverting an interval Hessian of a factorable function but Sisser's results have not yet been used to bound minimizers of factorable functions in a manner similar to that which has been discussed in the preceding sections using interval arithmetic.

Let $f : R^n \to R^1$ be a given twice continuously differentiable mapping and let $\nabla^2 f(\cdot) : R^n \to M(R^n)$ be the Hessian of f. If $\nabla^2 \underline{f}(\cdot) : I(R^n) \to I(M(R^n))$ is an interval extension of $\nabla^2 f(\cdot)$ then

$$\nabla^2 \underline{f}(\underline{x}) \supseteq \left\{ \nabla^2 f(x) \mid x \in \underline{x} \right\}. \tag{3.13.1}$$

Let $\underline{G}(\underline{x}) = \nabla^2 \underline{f}(\underline{x})$. We wish to compute $\underline{G}(\underline{x})^{-1} \in I(M(R^n))$ such that

$$\underline{G}(\underline{x})^{-1} = \left\{ \nabla^2 \underline{f}(\underline{x}) \right\}^{-1} \supseteq \left\{ G^{-1} \mid G \in \underline{G}(\underline{x}) \right\}.$$

ALGLIB [SheW--85] can provide computer-generated interval extensions of factorable functions and derivatives. We can write a program for the interval version of the procedure ***compute.sisser.functions*** in §2.3 to give an interval version of 2.3.2 according to

$$\nabla^2 \underline{f}(\underline{x}) = \sum_{i=1}^{m} \underline{\gamma}_i(\underline{x}) \left\{ \underline{a}_i(\underline{x}) \underline{b}_i(\underline{x})^T + \underline{b}_i(\underline{x}) \underline{a}_i(\underline{x})^T \right\}, \tag{3.13.2}$$

where for $(i = 1, \ldots, m)$ $\underline{\gamma}_i(\cdot) : I(R^n) \to I(R^1)$, $\underline{a}_i(\cdot) : I(R^n) \to I(R^n)$ and $\underline{b}_i(\cdot) : I(R^n) \to I(R^n)$. It is more convenient if 3.13.2 is rewritten in the symmetric form

$$\nabla^2 \underline{f}(\underline{x}) = \sum_{i=1}^{2m} \underline{s}_i(\underline{x}) \underline{u}_i(\underline{x}) \underline{u}_i(\underline{x})^T, \tag{3.13.3}$$

where

$$\underline{s}_i(\underline{x}) = \begin{cases} \frac{1}{2}\underline{\gamma}_i(\underline{x}) & (i = 1, \ldots, m), \\ -\frac{1}{2}\underline{\gamma}_{i-m}(\underline{x}) & (i = m+1, \ldots, 2m), \end{cases}$$

and

$$\underline{u}_i(\underline{x}) = \begin{cases} \underline{a}_i(\underline{x}) + \underline{b}_i(\underline{x}) & (i = 1, \ldots, m), \\ \underline{a}_{i-m}(\underline{x}) - \underline{b}_{i-m}(\underline{x}) & (i = m+1, \ldots, 2m). \end{cases}$$

The interval form of 3.2.7 is given by

$$\underline{G}(\underline{x}) = \underline{D} + \sum_{i=1}^{r} \underline{s}_i \underline{u}_i \underline{u}_i^T, \qquad 3.13.4$$

and

$$\underline{D} = \underline{\eta} \underline{I},$$

where $\underline{\eta} \in I(R)$ $(\eta_I > 0)$, $\underline{I}$ is the interval unit matrix, and $r = 2m + n$. In order to invert $\underline{G}$ which is given by 3.13.4 we have the following results.

Definition 3.13.1

The interval matrix $\underline{A} \in I(M(R^n))$ is nonsingular if and only if

$$0 \notin \left\{ det(A) \mid A \in \underline{A} \right\}. \square$$

Definition 3.13.2

An interval extension $\underline{det}(\cdot) : I(M(R^n)) \to I(R)$ of $det(\cdot) : M(R^n) \to R$ is such that

$$\underline{det}(\underline{A}) \supseteq \left\{ det(A) \mid A \in \underline{A} \right\}. \square$$

Theorem 3.13.1

Let $\underline{A} \in I(M(R^n))$ be nonsingular with $0 \notin \underline{det}(\underline{A})$. Let $\underline{s} \in I(R)$ and $\underline{x}, \underline{z} \in I(R^n)$ be given. If

$$[c, d] = 1 + \underline{s}\,\underline{z}^T \underline{A}^{-1} \underline{x} \qquad 3.13.5$$

and $0 \notin [c, d]$ then $\underline{A} + \underline{s}\,\underline{x}\,\underline{z}^T$ is nonsingular.

Proof

See [Sis---82c]. □

An interval version of the SMW formula is given in the following theorem.

Theorem 3.13.2

If (i) $\underline{A} \in I(M(R^n))$ and $0 \notin \underline{det}(\underline{A})$; (ii) $\underline{s} \in I(R)$ and $0 \notin \underline{s}$; (iii) $\underline{x}, \underline{z} \in I(R^n)$ and $0 \notin 1 + \underline{s}\,\underline{z}^T \underline{A}^{-1} \underline{x}$, then an interval extension $(\underline{A} + \underline{s}\,\underline{x}\,\underline{z}^T)^{-1}$ of $(A + sxz^T)^{-1}$ $(A \in \underline{A}, s \in \underline{s}, x \in \underline{x}$ and $z \in \underline{z})$ is given by

$$(\underline{A} + \underline{s}\,\underline{x}\,\underline{z}^T)^{-1} = \underline{A}^{-1} - \underline{A}^{-1}(\underline{s}^{-1} + \underline{z}^T \underline{A}^{-1} \underline{x})^{-1} \underline{x}\,\underline{z}^T \underline{A}^{-1} \qquad 3.13.6$$

Proof

See [Sis---82c]. □

The interval matrix $\underline{G}$ in 3.13.4 can be inverted using 3.13.6. However, it is difficult to know whether the resulting matrix is positive definite, because by using interval arithmetric, it might happen that in 3.13.5 $c \leq 0$ even though $\underline{A} + \underline{s}\,\underline{x}\,\underline{z}^T$ is positive definite. It really depends on how we compute the righthand sides of 3.13.5 and 3.13.6. Much work remains to be done on the use of Theorem 3.13.2.

Consider the problem of determining the zeros of $F : R^n \to R^n$. This problem can be solved by using the quadratic function $q : R^n \to R^1$ defined by [DenS--83]

$$q(x) = f(x^{(k)}) + g(x^{(k)})^T(x - x^{(k)}) +$$

$$\tfrac{1}{2}(x - x^{(k)})^T J(x^{(k)})^T J(x^{(k)})(x - x^{(k)}), \qquad 3.13.7$$

where the objective function $f : R^n \to R^1$ and the gradient $g : R^n \to R^n$ are given by

$$f(x^{(k)}) = \tfrac{1}{2}F(x^{(k)})^T F(x^{(k)}) \qquad 3.13.8$$

and

$$g(x^{(k)}) = J(x^{(k)})^T F(x^{(k)}) \qquad 3.13.9$$

recpectively. The Jacobian $J : R^n \to M(R^n)$ is given by 2.3.1. Let $s^{(k)} = x - x^{(k)}$, and let $J(x^{(k)})^T J(x^{(k)})$ be given by 2.3.24. Then 3.13.7 reduces to

$$q(x) = f(x^{(k)}) + g^{(k)^T} s^{(k)} + \tfrac{1}{2} s^{(k)^T} \left(\sum_{i=1}^{n} a_i(x^{(k)}) a_i(x^{(k)})^T \right) s^{(k)}$$

$$= f(x^{(k)}) + g^{(k)^T} s^{(k)} + \tfrac{1}{2} \sum_{i=1}^{n} (s^{(k)^T} a_i(x^{(k)}))^2, \qquad 3.13.10$$

where $a_i : R^n \to R^n$ $(i = 1, \ldots, n)$ are given by 2.3.25. The function q defined in 3.13.10 together with the minimization algorithms discussed in [DenS--83] can be used to solve the problem of determining the zeros of $F : R^n \to R^n$. On applying the $\mathcal{ALGLIB}$ package to this problem, we may create the data structures to represent the objective function

$$f(x) = \tfrac{1}{2}(F_1(x)^2 + \cdots + F_n(x)^2),$$

the Jacobian $J(x)$, $a_i(x) = \nabla F_i(x)$ $(i = 1, \ldots, n)$ and the gradient $g(x)$ using the appropriate $\mathcal{ALGLIB}$ procedures which are described in §2.2.

A further interesting and very important area of enquiry is concerned with the application of Differentiation Arithmetic [Ral---81] [Wol---87] to unconstrained optimization, but lack of time has prevented work on this area from being carried out.

CHAPTER 4

Procedures for Simultaneously Estimating and Bounding Simple Polynomial Zeros

4.1 Introduction

Several point iterative procedures for the simultaneous estimation of simple polynomial zeros exists; see for example, [Abe---73], [AleH--74], [BraH--73], [Ehr---67], [FarL--75], [HaPR--77], [Hen---74], [Ker---66], [MilP--83], [PetM--83], [PetS--86], [PetS--87] and references therein. Point iterative procedures can be very effective but have some disadvantages. For example, the known sufficient conditions for local convergence are usually difficult or impossible to verify computationally because they often involve knowledge of the zeros themselves ***a priori***; see §4.2. Also the sequences which are generated from point iterative procedures usually converge only for very good initial estimates of the zeros. Furthermore computationally rigorous bounds on the zeros are not obtained.

Several interval iterative procedures for the simultaneous inclusion of simple polynomial zeros also exist; see, for example, [Gar---75], [Gar---76], [Gar---78], [Gar---81], [GarH--72], [Gla---75], [Hen---74], [KriS--75], [MilP--83], [Pet---80], [Pet---82], [PetM--83], [PetS--85], [PetS--86]. Interval iterative procedures for the simultaneous inclusion of simple complex polynomial zeros determine bounded closed convex sets in C (usually rectangular or circular intervals) each of which contains a polynomial zero. If rectangular or circular machine interval arithmetic [AleH--83] is used, then the resulting intervals contain the exact polynomial zeros. Furthermore the widths of intervals are limited only by the precision of the machine floating point arithmetic. Thus interval iterative procedures can be used to determined very narrow computationally rigorous bounds on polynomial zeros.

If it is known that the zeros of a given polynomial are real then real interval arithmetic can be used and the partial ordering on the real numbers permits modifications for accelerating convergence to be introduced; this is exemplified in the book by Alefeld and Herzberger [AleH--83] in which the so-called interval total-step (IT) and interval single-step (IS) procedures are used to bound the eigenvalues of real symmetric matrices.

The purpose of this chapter is to describe the point repeated symmetric single-step procedure PRSS1 for simultaneously estimating simple polynomial zeros, and the interval repeated symmetric single-step procedure IRSS1 for simultaneously bounding simple polynomial zeros. The procedures PRSS1 and IRSS1 are based on the symmetric single-step idea of Aitken [Ait---50] and Alefeld [Ale---77].

The remainder of this chapter is organized as follows. Section 4.2 contains a brief survey of some of the existing point total-step and single-step procedures for simultaneously estimating simple polynomial zeros, and the new point procedure PRSS1 is described in Section 4.3. Section 4.4 contains a brief survey of some of the existing interval total-step and single-step procedures for simultaneously bounding simple polynomial zeros and the new interval procedure IRSS1 is described in Section 4.5. Iteration k of the procedure IRSS1 contains $m^{(k)}$ inner iterations. It is shown in Section 4.6 how $m^{(k)}$ may be determined automatically by the computer. Section 4.7 contains numerical results which illustrate the effectiveness of IRSS1.

4.2 Point Total-step and Single-step Procedures

Let $p : C^1 \rightarrow C^1$ be a polynomial of degree n defined by

$$p(x) = \sum_{i=0}^{n} a_i x^i \qquad 4.2.1$$

where $a_i \in C^1$ $(i = 0, \ldots, n)$ are given. This section contains several point iterative procedures for estimating the n simple zeros x_i^* $(i = 1, \ldots, n)$ of p simultaneously.

The equation

$$p(x) = 0 \tag{4.2.2}$$

can be expressed in the form

$$\prod_{j=1}^{n}(x - x_j^*) = 0 \tag{4.2.3}$$

if $a_n \neq 0$. Therefore it is assumed henceforth that $a_n = 1$, so that

$$p(x) = \prod_{j=1}^{n}(x - x_j^*). \tag{4.2.4}$$

Suppose that, for $j = 1, \ldots, n$, x_j is an estimate of x_j^*, and let $q : C^1 \to C^1$ be defined by

$$q(x) = \prod_{j=1}^{n}(x - x_j). \tag{4.2.5}$$

Then

$$q'(x_i) = \prod_{j \neq i}(x_i - x_j) \quad (i = 1, \ldots, n). \tag{4.2.6}$$

By 4.2.4, if, for $i = 1, \ldots, n$, $x_i \neq x_j^*$ $(j = 1, \ldots, n; j \neq i)$, then

$$x_i^* = x_i - \frac{p(x_i)}{\prod_{j \neq i}(x_i - x_j^*)}. \qquad 4.2.7$$

Now $x_j \approx x_j^*$ $(j = 1, \ldots, n)$ so by 4.2.7,

$$x_i^* \approx x_i - \frac{p(x_i)}{\prod_{j \neq i}(x_i - x_j)} \quad (i = 1, \ldots, n). \qquad 4.2.8$$

This gives rise to the point total-step procedure PT1 defined by

$$x_i^{(k+1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j \neq i}(x_i^{(k)} - x_j^{(k)})} \quad (i = 1, \ldots, n)\,(k \geq 0), \qquad 4.2.9$$

which has been studied by Weierstrass [Wei---03], Durand [Dur---60],and Kerner [Ker---66], and to the point single-step procedure PS1 defined by

$$x_i^{(k+1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k+1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k)})} \qquad 4.2.10$$

$$(i = 1, \ldots, n)\,(k \geq 0),$$

which has been studied by Alefeld and Herzberger [AleH--74].

The R-order of convergence of an iterative procedure is used in this thesis as a measure of the asymptotic convergence rate of the procedure. The concept of R-order of convergence is discussed in detail in [OrtR--70] and is discussed in a form which is sufficient for this thesis in [AleH--83]. The R-order of a procedure P which generates sequences which converge to z^* is denoted by $O_R(\mathrm{P}, z^*)$ and the R-factor of a sequence $\left(z^{(k)}\right)$ is denoted by $R_\nu(z^{(k)})$.

Theorem 4.2.1

If (1) $p : C^1 \to C^1$ defined by 4.2.1 has n distinct zeros x_i^* $(i = 1, \ldots, n)$; (2) $|x_i^{(0)} - x_i^*| \leq \theta d/(2n-1)$ $(i = 1, \ldots, n)$ where $0 < \theta < 1$ and $d = \min\{|x_i^* - x_j^*| \mid 1 \leq i < j \leq n\}$; (3) the sequences $\left(x_i^{(k)}\right)$ $(i = 1, \ldots, n)$ are generated from PT1 (*i.e.* from 4.2.9) then $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$ and $O_R(\text{PT1}, x_i^*) \geq 2$ $(i = 1, \ldots, n)$.

Proof

For $i = 1, \ldots, n$, let

$$|w_i^{(0)}| = |x_i^{(0)} - x_i^*|$$

$$\leq \theta d/(2n-1), \qquad 4.2.11$$

and in 4.2.9 let $x_i = x_i^{(k)}$, $\tilde{x}_i = x_i^{(k+1)}$. Then by 4.2.9

$$\tilde{x}_i = x_i - \frac{p(x_i)}{\prod_{j \neq i}(x_i - x_j)}$$

$$= x_i - \frac{\prod_{j=1}^{n}(x_i - x_j^*)}{\prod_{j \neq i}(x_i - x_j)} \qquad (i = 1, \ldots, n). \qquad 4.2.12$$

Let

$$q_i(x) = \prod_{m \neq i}(x - x_m). \qquad 4.2.13$$

Then

$$q_i'(x) = \sum_{l \neq i} \prod_{m \neq l,i} (x - x_m), \qquad 4.2.14$$

whence

$$q_i'(x_j) = \prod_{m \neq j,i} (x_j - x_m). \qquad 4.2.15$$

Let

$$\alpha_{ij} = \frac{1}{(x_j - x_i) q_i'(x_j)} \prod_{l \neq i,j} (x_j - x_l^*)$$

$$= \frac{\prod_{l \neq i,j} (x_j - x_l^*)}{(x_j - x_i) \prod_{m \neq j,i} (x_j - x_m)}$$

$$= \frac{\prod_{l \neq i,j} (x_j - x_l^*)}{\prod_{l \neq j} (x_j - x_l)}. \qquad 4.2.16$$

Then

$$\sum_{j \neq i} \alpha_{ij} (x_j - x_j^*) = \sum_{j \neq i} \frac{\prod_{l \neq i} (x_j - x_l^*)}{\prod_{l \neq j} (x_j - x_l)}. \qquad 4.2.17$$

Now by 4.2.12

$$\tilde{w} = \tilde{x}_i - x_i^*$$

$$= x_i - x_i^* - \frac{\prod_{l=1}^{n}(x_i - x_l^*)}{\prod_{l \neq i}(x_i - x_l)}$$

$$= (x_i - x_i^*)\left\{1 - \frac{\prod_{l \neq i}(x_i - x_l^*)}{\prod_{l \neq i}(x_i - x_l)}\right\}. \qquad 4.2.18$$

It can be shown (see Lemma 4.3.3) that for $i = 1, \ldots, n$

$$\sum_j \frac{\prod_{l \neq i}(x_j - x_l^*)}{\prod_{l \neq j}(x_j - x_l)} = 1. \qquad 4.2.19$$

So by 4.2.17, 4.2.18

$$\tilde{w}_i = w_i \sum_{j \neq i} \alpha_{ij} w_j \qquad (i = 1, \ldots, n). \qquad 4.2.20$$

Now by Hypothesis (2) and 4.2.11

$$|x_j - x_l^*| = |x_j^* - x_l^* - x_j^* + x_j|$$

$$\geq |x_j^* - x_l^*| - |x_j - x_j^*|$$

$$> d - \frac{1}{(2n-1)}d$$

$$= \frac{(2n-2)}{(2n-1)}d \qquad (j, l = 1, \ldots, n). \qquad 4.2.21$$

So by 4.2.21

$$|x_j - x_l| = |x_j - x_l^* + x_l^* - x_l|$$

$$\geq |x_j - x_l^*| - |x_l^* - x_l|$$

$$> \frac{(2n-2)}{(2n-1)}d - \frac{1}{(2n-1)}d$$

$$= \frac{(2n-3)}{(2n-1)}d \qquad (j,l = 1,\ldots,n). \tag{4.2.22}$$

So by 4.2.22

$$\frac{|x_j - x_l^*|}{|x_j - x_l|} = \frac{|x_j - x_l + x_l - x_l^*|}{|x_j - x_l|}$$

$$\leq \frac{|x_j - x_l| + |x_l - x_l^*|}{|x_j - x_l|}$$

$$= 1 + \frac{|x_l - x_l^*|}{|x_j - x_l|}$$

$$\leq 1 + \frac{d/(2n-1)}{(2n-3)d/(2n-1)}$$

$$= 1 + \frac{1}{2n-3}. \tag{4.2.23}$$

So by 4.2.16, 4.2.22, 4.2.23,

$$|\alpha_{ij}| = \frac{1}{|x_j - x_i|} \prod_{l \neq i,j} \frac{|x_j - x_l^*|}{|x_j - x_l|}$$

$$\leq \frac{(2n-1)}{(2n-3)} \frac{1}{d} \left\{ 1 + \frac{1}{2n-3} \right\}^{n-2} \qquad (i, j = 1, \ldots, n). \qquad 4.2.24$$

It can be shown that (see Lemma 4.3.1) ($\forall n \geq 2$)

$$\left\{ 1 + \frac{1}{2n-3} \right\}^{n-2} \leq \frac{(2n-3)}{(n-1)}. \qquad 4.2.25$$

So by 4.2.24

$$\frac{d}{(2n-1)} |\alpha_{ij}| \leq \frac{1}{(n-1)} \qquad (i, j = 1, \ldots, n). \qquad 4.2.26$$

Let

$$h_i = \frac{(2n-1)}{d} |w_i| \qquad (i = 1, \ldots, n), \qquad 4.2.27$$

and

$$\tilde{h}_i = \frac{(2n-1)}{d} |\tilde{w}_i| \qquad (i = 1, \ldots, n). \qquad 4.2.28$$

Then by 4.2.20

$$\tilde{h}_i \leq \frac{1}{(n-1)} h_i \sum_{j \neq i} h_j \qquad (i = 1, \ldots, n). \qquad 4.2.29$$

Now if

$$|w_i^{(0)}| \leq \frac{\theta}{(2n-1)}d \qquad (i = 1, \ldots, n), \tag{4.2.30}$$

where $0 < \theta < 1$, then by 4.2.27, $h_i \leq \theta$ $(i = 1, \ldots, n)$ whence by 4.2.29

$$\tilde{h}_i \leq \theta h_i \qquad (i = 1, \ldots, n). \tag{4.2.31}$$

Therefore by induction on k,

$$h_i^{(k)} \leq \theta^k h_i^{(0)} \qquad (i = 1, \ldots, n)\,(k \geq 0), \tag{4.2.32}$$

whence $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$. Let

$$h^{(k)} = \max_{1 \leq i \leq n} \{h_i^{(k)}\}.$$

Then by 4.2.29

$$h_i^{(k+1)} \leq \frac{1}{(n-1)} h_i^{(k)} \sum_{j \neq i} h_j^{(k)}$$

$$\leq \frac{1}{(n-1)} h^{(k)} (n-1) h^{(k)}$$

$$= h^{(k)2} \qquad (i = 1, \ldots, n),$$

whence

$$h_i^{(k+1)} \leq h^{(k)2} \qquad (i = 1, \ldots, n)(\forall k \geq 0).$$

So

$$O_R(\mathrm{PT1}, x^*) \geq 2. \quad \square$$

Theorem 4.2.2

If Hypotheses (1) and (2) of Theorem 4.2.1 are valid; (3) the sequences $\left(x_i^{(k)}\right)$ $(i = 1, \ldots, n)$ are generated from PS1 (*i.e.* from 4.2.10), then $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$, and $O_R(\mathrm{PS1}, x_i^*) \geq 1 + \tau$ $(i = 1, \ldots, n)$, where $\tau \in (1, 2)$ is the unique positive zero of $t^n - t - 1$.

Proof

Let the sequences $\left(x_i^{(k)}\right)$ $(i = 1, \ldots, n)$ be generated from 4.2.10. In 4.2.10 let $x_i = x_i^{(k)}$ and $\tilde{x}_i = x_i^{(k+1)}$. Then by 4.2.4 and 4.2.10

$$\tilde{x}_i = x_i - \frac{\prod_{j=1}^{n}(x_i - x_j^*)}{\prod_{j=1}^{i-1}(x_i - \tilde{x}_j)\prod_{j=i+1}^{n}(x_i - x_j)}. \qquad 4.2.33$$

Let

$$q_i(x) = \prod_{m=1}^{i-1}(x - \tilde{x}_m)\prod_{m=i+1}^{n}(x - x_m). \qquad 4.2.34$$

Then

$$q_i'(x) = \frac{d}{dx}\left\{\prod_{m=1}^{i-1}(x - \tilde{x}_m)\right\}\prod_{m=i+1}^{n}(x - x_m) + \prod_{m=1}^{i-1}(x - \tilde{x}_m)\frac{d}{dx}\left\{\prod_{m=i+1}^{n}(x - x_m)\right\}.$$

So

$$q_i'(\tilde{x}_j) = \prod_{\substack{m=1 \\ m\neq j}}^{i-1} (\tilde{x}_j - \tilde{x}_m) \prod_{m=i+1}^{n} (\tilde{x}_j - x_m) \quad (1 \leq j \leq i-1), \tag{4.2.35}$$

and

$$q_i'(x_j) = \prod_{m=1}^{i-1} (x_j - \tilde{x}_m) \prod_{\substack{m=i+1 \\ m\neq j}}^{n} (x_j - x_m) \quad (i+1 \leq j \leq n). \tag{4.2.36}$$

Let

$$\alpha_{ij} = \frac{\prod_{l\neq i,j} (\tilde{x}_j - x_l^*)}{(\tilde{x}_j - x_i)\, q_i'(\tilde{x}_j)}, \tag{4.2.37}$$

and

$$\beta_{ij} = \frac{\prod_{l\neq i,j} (x_j - x_l^*)}{(x_j - x_i)\, q_i'(x_j)}. \tag{4.2.38}$$

Then by 4.2.35 and 4.2.37

$$\alpha_{ij} = \frac{\prod_{l\neq i,j} (\tilde{x}_j - x_l^*)}{\prod_{\substack{m=1 \\ m\neq j}}^{i-1} (\tilde{x}_j - \tilde{x}_m) \prod_{m=i}^{n} (\tilde{x}_j - x_m)} \quad (1 \leq j \leq i-1) \tag{4.2.39}$$

and by 4.2.36 and 4.2.38

$$\beta_{ij} = \frac{\prod_{l\neq i,j} (x_j - x_l^*)}{\prod_{m=1}^{i-1} (x_j - \tilde{x}_m) \prod_{\substack{m=i \\ m\neq j}}^{n} (x_j - x_m)} \quad (i+1 \leq j \leq n). \tag{4.2.40}$$

It can be shown as in §4.3 that, if $w_i = x_i - x_i^*$ and $\tilde{w}_i = \tilde{x}_i - x_i^*$ for $i = 1, \ldots, n$, then

$$\tilde{w}_i = w_i \Big\{ \sum_{j=1}^{i-1} \alpha_{ij} \tilde{w}_j + \sum_{j=i+1}^{n} \beta_{ij} w_j \Big\} \quad (i = 1, \ldots, n). \tag{4.2.41}$$

By 4.2.40

$$|\beta_{1j}| = \frac{\prod_{l \neq 1,j} |x_j - x_l^*|}{\prod_{l \neq j} |x_j - x_l|}$$

$$= \frac{1}{|x_j - x_1|} \prod_{l \neq 1,j} \frac{|x_j - x_l^*|}{|x_j - x_l|} \quad (j = 2, \ldots, n). \tag{4.2.42}$$

So if 4.2.11 holds then

$$|\beta_{1j}| \leq \frac{(2n-1)}{d} \frac{1}{(n-1)} \quad (j = 2, \ldots, n). \tag{4.2.43}$$

So by 4.2.41

$$|\tilde{w}_1| \leq \frac{(2n-1)}{d} \frac{1}{(n-1)} |w_1| \sum_{j=2}^{n} |w_j|. \tag{4.2.44}$$

By 4.2.11

$$\frac{(2n-1)}{d} |w_j| < 1 \qquad (j = 1, \ldots, n).$$

So by 4.2.44

$$|\tilde{w}_1| \leq |w_1| \frac{1}{(n-1)} \sum_{j=2}^{n} \frac{(2n-1)}{d} |w_j|$$

$$< |w_1|.$$

Suppose that, for some $i \geq 2$, $|\tilde{w}_l| < |w_l|$ $(l = 1, \ldots, i-1)$. By 4.2.39

$$|\alpha_{ij}| = \frac{\prod_{l \neq i,j} |\tilde{x}_j - x_l^*|}{\prod_{\substack{l=1 \\ l \neq j}}^{i-1} |\tilde{x}_j - \tilde{x}_l| \prod_{l=i}^{n} |\tilde{x}_j - x_l|} \qquad (1 \leq j \leq i-1)$$

$$= \prod_{\substack{l=1 \\ l \neq j}}^{i-1} \frac{|\tilde{x}_j - x_l^*|}{|\tilde{x}_j - \tilde{x}_l|} \prod_{l=i+1}^{n} \frac{|\tilde{x}_j - x_l^*|}{|\tilde{x}_j - x_l|} \frac{1}{|\tilde{x}_j - x_i|}. \qquad 4.2.45$$

Now

$$\frac{|\tilde{x}_j - x_l^*|}{|\tilde{x}_j - \tilde{x}_l|} = \frac{|\tilde{x}_j - \tilde{x}_l + \tilde{x}_l - x_l^*|}{|\tilde{x}_j - \tilde{x}_l|}$$

$$\leq 1 + \frac{|\tilde{x}_l - x_l^*|}{|\tilde{x}_j - \tilde{x}_l|}. \qquad 4.2.46$$

Also

$$|\tilde{x}_j - x_l^*| = |x_j^* - x_l^* - x_j^* + \tilde{x}_j|$$

$$\geq |x_j^* - x_l^*| - |\tilde{x}_j - x_j^*|$$

$$> \frac{(2n-2)}{(2n-1)} d \quad (j = 1, \ldots, i-1; l = 1, \ldots, n), \qquad 4.2.47$$

because $|\tilde{w}_l| = |\tilde{x}_l - x_l^*| < |x_l - x_l^*| = |w_l|$ $(l = 1, \ldots, i-1)$. So

$$|\tilde{x}_j - \tilde{x}_l| = |\tilde{x}_j - x_l^* + x_l^* - \tilde{x}_l|$$

$$\geq |\tilde{x}_j - x_l^*| - |\tilde{x}_l - x_l^*|$$

$$> \frac{(2n-2)}{(2n-1)}d - \frac{1}{(2n-1)}d$$

$$= \frac{(2n-3)}{(2n-1)}d \quad (j,l = 1,\ldots,i-1). \qquad 4.2.48$$

Also

$$\frac{|\tilde{x}_j - x_l^*|}{|\tilde{x}_j - x_l|} = \frac{|\tilde{x}_j - x_l + x_l - x_l^*|}{|\tilde{x}_j - x_l|}$$

$$\leq 1 + \frac{|x_l - x_l^*|}{|\tilde{x}_j - x_l|}, \qquad 4.2.49$$

and so

$$|\tilde{x}_j - x_l| = |\tilde{x}_j - x_l^* + x_l^* - x_l|$$

$$\geq |\tilde{x}_j - x_l^*| - |x_l - x_l^*|$$

$$> \frac{(2n-2)}{(2n-1)}d - \frac{1}{(2n-1)}d$$

$$= \frac{(2n-3)}{(2n-1)}d \quad (j = 1,\ldots,i-1; l = 1,\ldots,n). \qquad 4.2.50$$

By 4.2.46–4.2.48

$$\prod_{\substack{l=1\\ l\neq j}}^{i-1} \frac{|\tilde{x}_j - x_l^*|}{|\tilde{x}_j - \tilde{x}_l|} \leq \prod_{\substack{l=1\\ l\neq j}}^{i-1} \left\{1 + \frac{1}{(2n-3)}\right\}$$

$$= \left\{1 + \frac{1}{(2n-3)}\right\}^{i-2}. \qquad 4.2.51$$

By 4.2.49, 4.2.50

$$\prod_{l=i+1}^{n} \frac{|\tilde{x}_j - x_l^*|}{|\tilde{x}_j - x_l|} \leq \prod_{l=i+1}^{n} \left\{1 + \frac{1}{2n-3}\right\}$$

$$= \left\{1 + \frac{1}{(2n-3)}\right\}^{n-i}. \qquad 4.2.52$$

So by 4.2.45, 4.2.50–4.2.52

$$|\alpha_{ij}| < \frac{(2n-1)}{(2n-3)} \frac{1}{d} \left\{1 + \frac{1}{2n-3}\right\}^{n-2} \quad (j = 1, \ldots, i-1; i = 2, \ldots, n),$$

whence by 4.2.49

$$\frac{d}{2n-1}|\alpha_{ij}| \leq \frac{1}{n-1} \quad (j = 1, \ldots, i-1; i = 2, \ldots, n). \qquad 4.2.53$$

Similarly,

$$\frac{d}{2n-1}|\beta_{ij}| \leq \frac{1}{n-1} \quad (j = i+1, \ldots, n; i = 2, \ldots, n). \qquad 4.2.54$$

So by 4.2.41 $|\tilde{w}_i| < |w_i|$. So by finite induction on i, $|\tilde{w}_i| < |w_i|$ $(i = 1, \ldots, n)$. As in the proof of Theorem 4.2.1, if 4.2.27 and 4.2.28 hold then 4.2.29 holds and $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$.

Let

$$h_i^{(k)} = \frac{(2n-1)}{d}|w_i^{(k)}| \quad (i = 1, \ldots, n)(k \geq 0). \tag{4.2.55}$$

If 4.2.11 holds then by 4.2.53, 4.2.54, 4.2.55, 4.2.41

$$h_i^{(k+1)} \leq \frac{1}{n-1} h_i^{(k)} \left\{ \sum_{j=1}^{i-1} h_j^{(k+1)} + \sum_{j=i+1}^{n} h_j^{(k)} \right\} \quad (i = 1, \ldots, n)(k \geq 0). \tag{4.2.56}$$

Suppose, without loss of generality, that

$$h_i^{(0)} \leq h < 1 \quad (i = 1, \ldots, n). \tag{4.2.57}$$

Then it follows that if

$$u_i^{(0)} = 1 \qquad (i = 1, \ldots, n),$$

$$A = \begin{pmatrix} 1 & 1 & & & & \\ & 1 & 1 & & \bigcirc & \\ & & \ddots & \ddots & & \\ & \bigcirc & & \ddots & 1 & \\ & & & & 1 & 1 \\ 1 & 1 & & & & 1 \end{pmatrix},$$

and

$$u^{(k+1)} = A\, u^{(k)} \qquad (k \geq 0),$$

then, as in the proof of Theorem 2 of [AleH--83]

$$h_i^{(k)} \leq h^{u_i^{(k)}} \qquad (i = 1, \ldots, n)(k \geq 0),$$

and

$$O_R(\mathrm{PS1}, x^*) \geq 1 + \sigma,$$

where $\sigma \in (1, 2)$ is the unique positive zero of $\tau^n - \tau - 1$. □

A derivation of the lower bound on the R-order of convergence for PT1 has been given by Ehrlich [Ehr---67] and outline of the derivation of the the lower bound on the R-order of convergence for PS1 has been given by Alefeld and Herzberger [AleH--74]. More detail on calculating lower bounds on R-orders of convergence is to be found in [AleH--83].

Let $\delta : C^1 \to C^1$ be defined by

$$\delta(x) = p(x)/q'(x) \qquad 4.2.58$$

where $p : C^1 \to C^1$ and $q : C^1 \to C^1$ are defined by 4.2.4 and 4.2.5 respectively. Then by 4.2.6 and 4.2.8, for $i = 1, \ldots, n$,

$$x_i^* \approx x_i - \delta(x_i). \qquad 4.2.59$$

Nourein [Nou---77a] has used 4.2.59 to approximate x_j^* in 4.2.7 to obtain the point total-step procedure PT2 defined by

$$x_i^{(k+1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j \neq i} (x_i^{(k)} - x_j^{(k)} + \delta(x_j^{(k)}))} \qquad (i = 1, \ldots, n)\,(k \geq 0). \qquad 4.2.60$$

The procedures PS1 and PT2 have been combined by Petković and Milovanović [PetM--83]

to give the point single-step procedure PS2 defined by

$$x_i^{(k+1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k+1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k)} + \delta(x_j^{(k)}))} \qquad 4.2.61$$

$$(i = 1, \ldots, n)\,(k \geq 0).$$

The procedure PS2 requires no more computational labour per iteration than does PT2, and the R-order of convergence [OrtR--70] of PS2 is at least $1 + \sigma$ where $2 \leq \sigma \leq 3$ [PetM--83] while PT2 exhibits cubic convergence [Nou---77a].

By 4.2.4, if $x_i \neq x_j^*$ $(i, j = 1, \ldots, n; i \neq j)$ then

$$\frac{p'(x_i)}{p(x_i)} = \sum_{j=1}^{n} \frac{1}{(x_i - x_j^*)}. \qquad 4.2.62$$

Let $\Delta : C^1 \to C^1$ be defined by

$$\Delta(x) = p(x)/p'(x). \qquad 4.2.63$$

Then by 4.2.62, for $i = 1, \ldots, n$,

$$x_i^* = x_i - \frac{\Delta(x_i)}{\left\{1 - \Delta(x_i) \sum_{j \neq i} \frac{1}{(x_i - x_j^*)}\right\}}. \qquad 4.2.64$$

This gives rise to the point total-step procedure PT3 [Mae---54], [Bor---63],[Ehr---67],

[Abe---73] defined by

$$\Delta_i^{(k)} = \Delta(x_i^{(k)}) \quad (i = 1, \ldots, n), \tag{4.2.65a}$$

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\Delta_i^{(k)}}{\left\{1 - \Delta_i^{(k)} \sum_{j \neq i} \frac{1}{(x_i^{(k)} - x_j^{(k)})}\right\}} \quad (i = 1, \ldots, n)\,(k \geq 0), \tag{4.2.65b}$$

which exhibits cubic convergence. The corresponding single-step procedure PS3 [AleH--74] is defined by

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\Delta_i^{(k)}}{\left[1 - \Delta_i^{(k)} \left\{ \sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - x_j^{(k+1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - x_j^{(k)})} \right\}\right]} \tag{4.2.66}$$

$$(i = 1, \ldots, n)\,(k \geq 0),$$

and has R-order of convergence at least $2+\sigma$ where $\sigma > 1$ is the unique positive zero of $t^n - t - 2$ [AleH--74], and is therefore preferable to PT3 which requires the same computational labour per iteration.

Theorem 4.2.3

If hypotheses (1) and (2) of Theorem 4.2.1 are valid; (3) the sequences $\left(x_i^{(k)}\right)$ $(i = 1, \ldots, n)$ are generated from PT3 (*i.e.* from 4.2.65) then $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$ and $O_R(\text{PT3}, x_i^*) \geq 3$ $(i = 1, \ldots, n)$. □

Theorem 4.2.4

If hypotheses (1) and (2) of Theorem 4.2.1 are valid; (3) the sequences $\left(x_i^{(k)}\right)$ $(i =$

$1, \ldots, n$) are generated from PS3 (*i.e.* from 4.2.66) then $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$ and $O_R(\text{PS3}, x_i^*) \geq 2 + \tau$ $(i = 1, \ldots, n)$, where $\tau \in (1, 2]$ is the unique positive zero of $t^n - t - 2$. □

The convergence proofs for these theorems are similar to that of Theorem 4.2.3. The derivations of the lower bounds on the R-orders of convergence are given in [Ehr---67] and in [AleH--74] respectively.

The procedure PT3 has been modified by Nourein [Nou---77a] to obtain the point total-step procedure PT4 defined by

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\Delta_i^{(k)}}{\left\{1 - \Delta_i^{(k)} \sum_{j \neq i} \frac{1}{(x_i^{(k)} - x_j^{(k)} + \Delta_j^{(k)})}\right\}} \qquad 4.2.67$$

$$(i = 1, \ldots, n)\,(k \geq 0).$$

The procedure PT4 has order of convergence 4 and requires no more computational labour per iteration than does PT3 and PS3, and is therefore preferable to PT3 and PS3.

It has been pointed out by Petković and Milovanović [PetM--83] that the single-step modification PS4 of PT4 defined by

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\Delta_i^{(k)}}{\left[1 - \Delta_i^{(k)} \left\{\sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - x_j^{(k+1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - x_j^{(k)} + \Delta_j^{(k)})}\right\}\right]}$$

$$(i = 1, \ldots, n)\,(k \geq 0), \qquad 4.2.68$$

is more rapidly convergent than PT4 while requiring no more computational labour per iteration. The R-order of convergence of PS4 is at least $2(1+\tau)$ where $\tau \in (1,2)$ is the unique positive zero of $t^n - t - 1$ [MilP--83], so that PS4 is preferable to PT4.

Theorem 4.2.5

If hypotheses (1) and (2) of Theorem 4.2.1 are valid; (3) the sequences $\left(x_i^{(k)}\right)$ $(i = 1,\ldots,n)$ are generated from PT4 (*i.e.* from 4.2.67) then $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1,\ldots,n)$ and $O_R(\text{PT4}, x_i^*) \geq 4$ $(i = 1,\ldots,n)$. □

Theorem 4.2.6

If hypotheses (1) and (2) of Theorem 4.2.1 are valid; (3) the sequences $\left(x_i^{(k)}\right)$ $(i = 1,\ldots,n)$ are generated from PS4 (*i.e.* from 4.2.68), then $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1,\ldots,n)$ and $O_R(\text{PS4}, x_i^*) \geq 2(1+\tau)$ $(i = 1,\ldots,n)$, where $\tau \in (1,2)$ is the unique positive zero of $t^n - t - 1$. □

The convergence proofs for these theorems are similar to that of Theorem 4.2.4. The derivations of the lower bounds on the R-orders of convergence are given in [Nou---77a] and [PetM--83] respectively.

Let $\phi : C^1 \to C^1$ be defined by

$$\phi(x) = q(x) + \sum_{i=1}^{n} \frac{p(x_i)q(x)}{q'(x_i)(x - x_i)} \qquad 4.2.69$$

where $p : C^1 \to C^1$ and $q : C^1 \to C^1$ are defined by 4.2.4 and 4.2.5 respectively. Then for $k = 1,\ldots,n$,

$$\phi(x_k) = q(x_k) + \sum_{i=1}^{n} \frac{\prod_{j \neq i}(x_k - x_j)p(x_i)}{\prod_{j \neq i}(x_i - x_j)}. \qquad 4.2.70$$

Now by 4.2.5,

$$q(x_k) = 0 \quad (k = 1, \ldots, n). \qquad 4.2.71$$

Furthermore for $k = 1, \ldots, n$,

$$\frac{\prod_{j \neq i}(x_k - x_j)}{\prod_{j \neq i}(x_i - x_j)} = \begin{cases} 1 & (k = i) \\ 0 & (k \neq i) \end{cases}. \qquad 4.2.72$$

By 4.2.70– 4.2.72,

$$\phi(x_k) = p(x_k) \quad (k = 1, \ldots, n). \qquad 4.2.73$$

By 4.2.4 and 4.2.5, $q(x)/p(x) \to 1$ $(x \to \infty)$, so by 4.2.69, $\phi(x)/p(x) \to 1$ $(x \to \infty)$. The polynomial ϕ is of degree n and interpolates p at the n points x_i $(i = 1, \ldots, n)$ and the point at infinity. Therefore, by the uniqueness of the Lagrange interpolating polynomial, $\phi(x) = p(x)$ $(x \in C)$. Therefore by 4.2.69

$$p(x) = q(x) + \sum_{i=1}^{n} \frac{p(x_i)q(x)}{q'(x_i)(x - x_i)}. \qquad 4.2.74$$

Let $\delta : C^1 \to C^1$ be defined by 4.2.58. Then by 4.2.74, for $i = 1, \ldots, n$,

$$x_i^* = x_i - \frac{\delta(x_i)}{\left\{1 + \sum_{j \neq i} \frac{\delta(x_j)}{(x_i^* - x_j)}\right\}}, \qquad 4.2.75$$

because $p(x_i^*) = 0$ $(i = 1, \ldots, n)$. This gives rise to the point total-step procedure PT5 [Bor---70] defined by

$$\delta_i^{(k)} = \delta(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\delta_i^{(k)}}{\left\{1 + \sum_{j \neq i} \frac{\delta_j^{(k)}}{(x_i^{(k)} - x_j^{(k)})}\right\}} \quad (i = 1, \ldots, n)\,(k \geq 0). \qquad 4.2.76$$

The procedure PT5 is cubically convergent if for $i = 1, \ldots, n$, $x_i^{(0)}$ is sufficiently close to x_i^* and the x_i^* are simple zeros of p [Nou---75].

Nourein [Nou---77b] has modified PT5 to produce the point total-step procedure PT6 defined by

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\delta_i^{(k)}}{\left\{1 + \sum_{j \neq i} \frac{\delta_j^{(k)}}{(x_i^{(k)} - \delta_i^{(k)} - x_j^{(k)})}\right\}} \quad (i = 1, \ldots, n)\,(k \geq 0), \qquad 4.2.77$$

which has order of convergence 4, and requires no more computational labour per iteration than does PT5.

4.3 The Point Repeated Symmetric Single-step Procedure PRSS1

In this section the symmetric single-step idea of Aitken [Ait---50] is used to derive a point iterative procedure PRSS1 which appears to be new.

The symmetric point single-step procedure PSS1 corresponding to the point single-step procedure PS1 is defined by

$$x_i^{(k,0)} = x_i^{(k)} \quad (i = 1, \ldots, n), \tag{4.3.1a}$$

$$x_i^{(k,1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k,0)})} \quad (i = 1, \ldots, n), \tag{4.3.1b}$$

$$x_i^{(k,2)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k,2)})} \quad (i = n, \ldots, 1), \tag{4.3.1c}$$

$$x_i^{(k+1)} = x_i^{(k,2)} \quad (i = 1, \ldots, n), \ (k \geq 0). \tag{4.3.1d}$$

The procedure PSS1 has the following attractive features:

(*i*) The values $p(x_i^{(k)})$ $(i = 1, \ldots, n)$ which are computed for use in 4.3.1b are re-used in 4.3.1c.

(*ii*) The products

$$\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,1)}) \quad (i = 2, \ldots, n)$$

which are computed for use in 4.3.1b are re-used in 4.3.1c.

(*iii*) $x_n^{(k,1)} = x_n^{(k,2)}$ $(k \geq 0)$ so that $x_n^{(k,2)}$ need not be computed.

(*iv*) The R-order of convergence $O_R(\mathrm{PS1}, x^*)$ for PS1 to the set of simple zeros $x^* = \left(x_i^*\right)_{n\times 1}$ is such that

$$O_R(\mathrm{PS1}, x^*) \geq 1 + \tau > 2,$$

where τ is the unique positive zero of $t^n - t - 1$ [AleH--74]. As shown subsequently in this section, the corresponding R-order of convergence $O_R(\mathrm{PSS1}, x^*)$ is at least 3.

The value of $x_i^{(k,2)}$ which is computed from 4.3.1c requires $(n - i)$ multiplications, one division, and $(n - i + 1)$ subtractions, increasing the lower bound on the R-order by unity compared with the R-order of PS1. This observation gives rise to the idea that it might be advantageous to repeat the second step in PSS1 $r^{(k)}$ times in each iteration where the integer $r^{(k)} \geq 1$ depends on the iteration number k. This leads to the repeated symmetric single-step procedure PRSS1 which consists of generating the sequences $\left(x_i^{(k)}\right)$ $(i = 1, \ldots, n)$ from

$$x_i^{(k,0)} = x_i^{(k)} \quad (i = 1, \ldots, n), \tag{4.3.2a}$$

$$x_i^{(k,2l-1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k,2l-2)})} \quad (i = 1, \ldots, n), \tag{4.3.2b}$$

$$x_i^{(k,2l)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k,2l)})} \quad (i = n, \ldots, 1), \tag{4.3.2c}$$

$$\left(l = 1, \ldots, r^{(k)}\right),$$

$$x_i^{(k+1)} = x_i^{(k,2r^{(k)})} \quad (i = 1, \ldots, n)\,(k \geq 0). \tag{4.3.2d}$$

If $r^{(k)} = 1$ $(\forall k \geq 0)$ then PRSS1 and PSS1 coincide.

From 4.3.2*b* and 4.3.2*c*, it follows that for $l \geq 1$, $x_n^{(k,2l)} = x_n^{(k,2l-1)}$ and $x_1^{(k,2l+1)} = x_1^{(k,2l)}$ $(k \geq 0)$ so that $x_n^{(k,2l)}$ and $x_1^{(k,2l+1)}$ need not be computed.

The following lemmas are required in the proof of Theorem 4.3.1.

Lemma 4.3.1

$$\left(1+\frac{1}{2n-3}\right)^{n-2} \leq \frac{2n-3}{n-1} \quad (\forall n \geq 2).$$

Proof

Let

$$y(n) = log\left\{2^{n-2}\left(\frac{n-1}{2n-3}\right)^{n-1}\right\} \quad (n \geq 2).$$

Then $y(2) = 0$, $y'(2) = log\, 2 - 1 < 0$, and $y'(n) < 0$ $(\forall n \geq 3)$. So $y(n) \leq 0$ $(\forall n \geq 2)$, whence

$$1 \geq 2^{n-2}\left(\frac{n-1}{2n-3}\right)^{n-1}$$

$$= \frac{(n-1)}{(2n-3)}\left(1+\frac{1}{2n-3}\right)^{n-2} \quad (\forall n \geq 2).$$

This proves the lemma. □

Lemma 4.3.2

If (1) $p : C \to C$ is defined by 4.2.4; (2) $p_i : C \to C$ is defined by

$$p_i(x) = \prod_{m=1}^{i-1}(x - x_m^*) \prod_{m=i+1}^{n}(x - x_m^*) \quad (i = 1, \ldots, n); \tag{4.3.3}$$

(3) $q_i : C \to C$ is defined by

$$q_i(x) = \prod_{m=1}^{i-1}(x - \bar{x}_m) \prod_{m=i+1}^{n}(x - \hat{x}_m) \quad (i = 1, \ldots, n), \tag{4.3.4}$$

where $\bar{x}_j \neq \bar{x}_m$ and $\hat{x}_j \neq \hat{x}_m$ $(j, m = 1, \ldots, n; j \neq m)$; (4) $\phi_i : C \to C$ is defined by

$$\phi_i(x) = q_i(x) + \sum_{j=1}^{i-1} \frac{p_i(\bar{x}_j)q_i(x)}{q_i'(\bar{x}_j)(x - \bar{x}_j)} + \sum_{j=i+1}^{n} \frac{p_i(\hat{x}_j)q_i(x)}{q_i'(\hat{x}_j)(x - \hat{x}_j)} \quad (i = 1, \ldots, n), \tag{4.3.5}$$

then

$$\phi_i(x) = p_i(x) \quad (\forall x \in C)\,(i = 1, \ldots, n).$$

Proof

By 4.3.4, for $j = 1, \ldots, i - 1$,

$$\frac{q_i(x)}{(x - \bar{x}_j)} = \prod_{\substack{m=1 \\ m \neq j}}^{i-1}(x - \bar{x}_m) \prod_{m=i+1}^{n}(x - \hat{x}_m).$$

So by 4.3.4, for $j, k = 1, \ldots, i - 1$,

$$\frac{q_i(\bar{x}_k)}{q_i'(\bar{x}_j)(\bar{x}_k - \bar{x}_j)} = \prod_{\substack{m=1 \\ m \neq j}}^{i-1}\left(\frac{\bar{x}_k - \bar{x}_m}{\bar{x}_j - \bar{x}_m}\right) \prod_{m=i+1}^{n}\left(\frac{\bar{x}_k - \hat{x}_m}{\bar{x}_j - \hat{x}_m}\right)$$

$$= \begin{cases} 1 & (j = k) \\ \\ 0 & (j \neq k) \end{cases}.$$

Similarly for $j, k = i+1, \ldots, n$,

$$\frac{q_i(\hat{x}_k)}{q_i'(\hat{x}_j)(\hat{x}_k - \hat{x}_j)} = \prod_{m=1}^{i-1} \left(\frac{\hat{x}_k - \bar{x}_m}{\hat{x}_j - \bar{x}_m} \right) \prod_{\substack{m=i+1 \\ m \neq j}}^{n} \left(\frac{\hat{x}_k - \hat{x}_m}{\hat{x}_j - \hat{x}_m} \right)$$

$$= \begin{cases} 1 & (j = k) \\ \\ 0 & (j \neq k) \end{cases}.$$

Furthermore, by 4.3.4,

$$q_i(\hat{x}_k) = 0 \quad (i+1 \leq k \leq n),$$

and

$$q_i(\bar{x}_k) = 0 \quad (1 \leq k \leq i-1).$$

Therefore by 4.3.5,

$$\phi_i(\bar{x}_k) = p_i(\bar{x}_k) \quad (k = 1, \ldots, i-1),$$

and

$$\phi_i(\hat{x}_k) = p_i(\hat{x}_k) \quad (k = i+1, \ldots, n).$$

Finally, $\phi_i(x)/p_i(x) \to 1$ $(x \to \infty)$ $(i = 1, \ldots, n)$, so ϕ_i interpolates p_i at the $n-1$ points $\bar{x}_1, \ldots, \bar{x}_{i-1}, \hat{x}_{i+1}, \ldots, \hat{x}_n$, and the point at infinity. Therefore by the uniqueness of the La-

grange interpolating polynomial, $\phi_i(x) = p_i(x)$ $(\forall x \in C)$ $(i = 1, \ldots, n)$. □

Lemma 4.3.3

If hypotheses (1)–(4) of Lemma 4.3.2 are valid; (5) $\check{x}_i$ $(i = 1, \ldots, n)$ are such that $p(\check{x}_i) \neq 0$ $(i = 1, \ldots, n)$, $\check{x}_i \neq \bar{x}_m$ $(m = 1, \ldots, i-1)$, $\check{x}_i \neq \hat{x}_m$ $(m = i+1, \ldots, n)$, and

$$\bar{x}_i = \check{x}_i - \frac{p(\check{x}_i)}{\prod_{m=1}^{i-1}(\check{x}_i - \bar{x}_m) \prod_{m=i+1}^{n}(\check{x}_i - \hat{x}_m)} \quad (i = 1, \ldots, n); \tag{4.3.6}$$

(6) $\check{w}_i = \check{x}_i - x_i^*$, $\hat{w}_i = \hat{x}_i - x_i^*$, and $\bar{w}_i = \bar{x}_i - x_i^*$ $(i = 1, \ldots, n)$, then

$$\bar{w}_i = \check{w}_i \left\{ \sum_{j=1}^{i-1} \bar{\gamma}_{ij} \bar{w}_j + \sum_{j=i+1}^{n} \hat{\gamma}_{ij} \hat{w}_j \right\} \quad (i = 1, \ldots, n), \tag{4.3.7}$$

where

$$\bar{\gamma}_{ij} = \frac{\prod_{m \neq i,j} (\bar{x}_j - x_m^*)}{q_i'(\bar{x}_j)(\bar{x}_j - \check{x}_i)} \quad (j = 1, \ldots, i-1), \tag{4.3.8}$$

and

$$\hat{\gamma}_{ij} = \frac{\prod_{m \neq i,j} (\hat{x}_j - x_m^*)}{q_i'(\hat{x}_j)(\hat{x}_j - \check{x}_i)} \quad (j = i+1, \ldots, n). \tag{4.3.9}$$

Proof

By 4.3.4, $q_i(\check{x}_i) \neq 0$ $(i = 1, \ldots, n)$. So by 4.3.5, Lemma 4.3.2, 4.3.8, and 4.3.9

$$1 - \frac{p_i(\check{x}_i)}{q_i(\check{x}_i)} = \sum_{j=1}^{i-1} \frac{p_i(\bar{x}_j)}{q_i'(\bar{x}_j)(\bar{x}_j - \check{x}_i)} + \sum_{j=i+1}^{n} \frac{p_i(\hat{x}_j)}{q_i'(\hat{x}_j)(\hat{x}_j - \check{x}_i)}$$

$$= \sum_{j=1}^{i-1} \bar{\gamma}_{ij} \bar{w}_j + \sum_{j=i+1}^{n} \hat{\gamma}_{ij} \hat{w}_j. \qquad 4.3.10$$

Also, by 4.3.6

$$\bar{w}_i = \check{w}_i \left\{ 1 - \frac{p_i(\check{x}_i)}{q_i(\check{x}_i)} \right\},$$

whence 4.3.7 follows from 4.3.10. □

Lemma 4.3.4

If hypotheses (1)–(5) of Lemma 4.3.3 are valid; (6) $|\check{x}_i - x_i^*| \leq \theta d/(2n-1)$ and $|\hat{x}_i - x_i^*| \leq \theta d/(2n-1)$ $(i = 1, \ldots, n)$ where $d = \min\{|x_i^* - x_j^*| \mid i, j = 1, \ldots, n; i \neq j\}$ and $0 < \theta < 1$, then $|\bar{w}_i| \leq \theta |\check{w}_i|$ $(i = 1, \ldots, n)$.

Proof

Now

$$|\hat{x}_j - x_m^*| \geq |x_j^* - x_m^*| - |\hat{x}_j - x_j^*|$$

$$\geq d - \frac{\theta}{(2n-1)} d$$

$$\geq \left(\frac{2n-2}{2n-1} \right) d \quad (j, m = 1, \ldots, n),$$

whence

$$|\hat{x}_j - \hat{x}_m| \geq |\hat{x}_j - x_m^*| - |x_m^* - \hat{x}_m|$$

$$\geq \left(\frac{2n-2}{2n-1}\right) d - \frac{\theta}{(2n-1)} d$$

$$\geq \left(\frac{2n-3}{2n-1}\right) d \quad (j, m = 1, \ldots, n).$$

Therefore

$$\frac{|\hat{x}_j - x_m^*|}{|\hat{x}_j - \hat{x}_m|} \leq 1 + \frac{|\hat{x}_m - x_m^*|}{|\hat{x}_j - \hat{x}_m|}$$

$$\leq 1 + \frac{1}{(2n-3)} \quad (j, m = 1, \ldots, n).$$

Also

$$|\hat{x}_j - \check{x}_i| \geq |\hat{x}_j - x_i^*| - |x_i^* - \check{x}_i|$$

$$\geq \left(\frac{2n-3}{2n-1}\right) d \quad (i, j = 1, \ldots, n).$$

So by 4.3.9 and Lemma 4.3.1,

$$|\hat{\gamma}_{1j}| \leq \prod_{\substack{m=2 \\ m \neq j}}^{n} \left(1 + \frac{1}{2n-3}\right) \frac{(2n-1)}{(2n-3)d}$$

$$\leq \frac{1}{(n-1)} \frac{(2n-1)}{d} \quad (j = 2, \ldots, n).$$

So by 4.3.7,

$$|\bar{w}_1| \leq |\check{w}_1| \sum_{j=2}^{n} \frac{1}{(n-1)} \frac{(2n-1)}{d} |\hat{w}_j|$$

$$\leq \theta |\check{w}_1|.$$

Suppose that for some $i \geq 2$, $|\bar{w}_m| \leq \theta|\check{w}_m|$ $(m = 1, \ldots, i-1)$. Then

$$|\bar{x}_j - x_m^*| \geq |x_j^* - x_m^*| - |\bar{x}_j - x_j^*|$$

$$\geq d - \frac{\theta}{(2n-1)}d$$

$$\geq \left(\frac{2n-2}{2n-1}\right)d \quad (j = 1, \ldots, i-1; m = 1, \ldots, n).$$

So

$$|\bar{x}_j - \bar{x}_m| \geq |\bar{x}_j - x_m^*| - |\bar{x}_m - x_m^*|$$

$$\geq \left(\frac{2n-3}{2n-1}\right)d \quad (j, m = 1, \ldots, i-1),$$

whence

$$\frac{|\bar{x}_j - x_m^*|}{|\bar{x}_j - \bar{x}_m|} \leq 1 + \frac{|\bar{x}_m - x_m^*|}{|\bar{x}_j - \bar{x}_m|}$$

$$\leq 1 + \frac{1}{(2n-3)} \quad (j, m = 1, \ldots, i-1).$$

Similarly,

$$\frac{|\bar{x}_j - x_m^*|}{|\bar{x}_j - \hat{x}_m|} \leq 1 + \frac{1}{(2n-3)} \quad (j = 1, \ldots, i-1; m = 1, \ldots, n).$$

Also

$$|\bar{x}_j - \check{x}_i| \geq |\bar{x}_j - x_i^*| - |x_i^* - \check{x}_i|$$

$$\geq \left(\frac{2n-3}{2n-1}\right) d \quad (j = 1, \ldots, i-1; i = 1, \ldots, n).$$

So by 4.3.8, and Lemma 4.3.1,

$$|\bar{\gamma}_{ij}| \leq \frac{1}{(n-1)} \frac{(2n-1)}{d} \quad (j = 1, \ldots, i-1; i = 1, \ldots, n).$$

Similarly, by 4.3.9 and Lemma 4.3.1,

$$|\hat{\gamma}_{ij}| \leq \frac{1}{(n-1)} \frac{(2n-1)}{d} \quad (j = i+1, \ldots, n; i = 1, \ldots, n).$$

So by 4.3.7,

$$|\bar{w}_i| \leq |\check{w}_i| \frac{1}{(n-1)} \frac{(2n-1)}{d} \left\{ \sum_{j=1}^{i-1} |\bar{w}_j| + \sum_{j=i+1}^{n} |\hat{w}_j| \right\}$$

$$\leq \theta |\check{w}_i|.$$

So by finite induction on i, $|\bar{w}_i| \leq \theta |\check{w}_i|$ $(i = 1, \ldots, n)$. □

Theorem 4.3.1

If (1) $p : C \to C$ defined by 4.2.4 has n distinct zeros x_i^* $(i = 1, \ldots, n)$; (2) $|x_i^{(0)} - x_i^*| \leq \theta d/(2n-1)$ $(i = 1, \ldots, n)$ where $0 < \theta < 1$ and $d = \min\{|x_i^* - x_j^*| \mid i, j = 1, \ldots, n; i \neq j\}$; (3) the sequences $\left(x_i^{(k)}\right)$ $(i = 1, \ldots, n)$ are generated from PRSS1 (*i.e.* from 4.3.2) with $r^{(k)} = r \geq 1$ $(\forall k \geq 0)$, then $x_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$ and $O_R(\text{PRSS1}, x^*) \geq 2r + 1$.

Proof

For $l = 1, \ldots, r$, $i = 1, \ldots, n$, let

$$q_{2l-1,i}(x) = \prod_{m=1}^{i-1}(x - x_m^{(k,2l-1)}) \prod_{m=i+1}^{n}(x - x_m^{(k,2l-2)}), \tag{4.3.11}$$

$$q_{2l,i}(x) = \prod_{m=1}^{i-1}(x - x_m^{(k,2l-1)}) \prod_{m=i+1}^{n}(x - x_m^{(k,2l)}), \tag{4.3.12}$$

$$\phi_{2l-1,i}(x) = q_{2l-1,i}(x) + \sum_{j=1}^{i-1} \frac{p_i(x_j^{(k,2l-1)}) q_{2l-1,i}(x)}{q'_{2l-1,i}(x_j^{(k,2l-1)})(x - x_j^{(k,2l-1)})} +$$

$$\sum_{j=i+1}^{n} \frac{p_i(x_j^{(k,2l-2)}) q_{2l-1,i}(x)}{q'_{2l-1,i}(x_j^{(k,2l-2)})(x - x_j^{(k,2l-2)})}, \tag{4.3.13}$$

and

$$\phi_{2l,i}(x) = q_{2l,i}(x) + \sum_{j=1}^{i-1} \frac{p_i(x_j^{(k,2l-1)}) q_{2l,i}(x)}{q'_{2l,i}(x_j^{(k,2l-1)})(x - x_j^{(k,2l-1)})} +$$

$$\sum_{j=i+1}^{n} \frac{p_i(x_j^{(k,2l)}) q_{2l,i}(x)}{q'_{2l,i}(x_j^{(k,2l)})(x - x_j^{(k,2l)})}, \tag{4.3.14}$$

where $p_i(x)$ is defined by 4.3.3.

By Lemma 4.3.2 and Lemma 4.3.3 with $q_i = q_{2l-1,i}$, $\check{x}_i = x_i^{(k)}$, $\hat{x}_i = x_i^{(k,2l-2)}$, $\bar{x}_i = x_i^{(k,2l-1)}$, $\phi_i = \phi_{2l-1,i}$ $(i = 1, \ldots, n)$ $(l = 1, \ldots, r)$, it follows that for $i = 1, \ldots, n$, $l = 1, \ldots, r$,

$k \geq 0$,

$$w_i^{(k,2l-1)} = w_i^{(k)} \left\{ \sum_{j=1}^{i-1} \alpha_{ij}^{(k,2l-1)} w_j^{(k,2l-1)} + \sum_{j=i+1}^{n} \alpha_{ij}^{(k,2l-2)} w_j^{(k,2l-2)} \right\}, \qquad 4.3.15$$

where

$$w_i^{(k,s)} = x_i^{(k,s)} - x_i^* \quad (s = 0, \ldots, r),$$

$$\alpha_{ij}^{(k,2l-1)} = \frac{\prod_{m \neq i,j} (x_j^{(k,2l-1)} - x_m^*)}{q'_{2l-1,i}(x_j^{(k,2l-1)})(x_j^{(k,2l-1)} - x_i^{(k)})} \quad (j = 1, \ldots, i-1), \qquad 4.3.16$$

and

$$\alpha_{ij}^{(k,2l-2)} = \frac{\prod_{m \neq i,j} (x_j^{(k,2l-2)} - x_m^*)}{q'_{2l-1,i}(x_j^{(k,2l-2)})(x_j^{(k,2l-2)} - x_i^{(k)})} \quad (j = i+1, \ldots, n). \qquad 4.3.17$$

Similarly, by Lemma 4.3.2 and Lemma 4.3.3, with $q_i = q_{2l,i}$, $\check{x}_i = x_i^{(k)}$, $\bar{x}_i = x_i^{(k,2l-1)}$, $\hat{x}_i = x_i^{(k,2l)}$, $\phi_i = \phi_{2l,i}$ $(i = 1, \ldots, n)$ $(l = 1, \ldots, r)$, it follows that for $i = 1, \ldots, n$, $l = 1, \ldots, r$, $k \geq 0$,

$$w_i^{(k,2l)} = w_i^{(k)} \left\{ \sum_{j=1}^{i-1} \beta_{ij}^{(k,2l-1)} w_j^{(k,2l-1)} + \sum_{j=i+1}^{n} \beta_{ij}^{(k,2l)} w_j^{(k,2l)} \right\}, \qquad 4.3.18$$

where

$$\beta_{ij}^{(k,2l-1)} = \frac{\prod_{m \neq i,j} (x_j^{(k,2l-1)} - x_m^*)}{q'_{2l,i}(x_j^{(k,2l-1)})(x_j^{(k,2l-1)} - x_i^{(k)})} \quad (j = 1, \ldots, i-1), \qquad 4.3.19$$

and

$$\beta_{ij}^{(k,2l)} = \frac{\prod_{m \neq i,j} (x_j^{(k,2l)} - x_m^*)}{q'_{2l,i}(x_j^{(k,2l)})(x_j^{(k,2l)} - x_i^{(k)})} \quad (j = i+1, \ldots, n). \tag{4.3.20}$$

It follows from 4.3.15–4.3.17 and Lemma 4.3.4 that $|w_i^{(0,1)}| \leq \theta |w_i^{(0,0)}|$ $(i = 1, \ldots, n)$, and it follows from 4.3.18–4.3.20 and Lemma 4.3.4 that $|w_i^{(0,2)}| \leq \theta^2 |w_i^{(0,0)}|$ $(i = 1, \ldots, n)$. Suppose that for some $l \geq 1$,

$$|w_i^{(0,2l)}| \leq \theta^{2l} |w_i^{(0,0)}| \quad (i = 1, \ldots, n).$$

Then it follows from 4.3.15–4.3.20 by finite induction on l that

$$|w_i^{(0,2r)}| \leq \theta^{2r} |w_i^{(0,0)}| \quad (i = 1, \ldots, n),$$

whence $|w_i^{(1,0)}| \leq \theta^{2r} |w_i^{(0,0)}|$ $(i = 1, \ldots, n)$. It then follows by induction on k that $\forall k \geq 0$

$$|w_i^{(k,0)}| \leq \theta^{(2r+1)^k - 1} |w_i^{(0,0)}| \quad (i = 1, \ldots, n),$$

whence $x_i^{(k)} \to x_i^*$ $(k \to \infty)$, $(i = 1, \ldots, n)$. Let

$$h_i^{(k,m)} = \frac{(2n-1)}{d} |w_i^{(k,m)}| \quad (i = 1, \ldots, n)\,(m = 0, \ldots, 2r).$$

Then by 4.3.15 and 4.3.18, for $i = 1, \ldots, n$,

$$h_i^{(k,2l-1)} \leq \frac{1}{(n-1)} h_i^{(k,0)} \left\{ \sum_{j=1}^{i-1} h_j^{(k,2l-1)} + \sum_{j=i+1}^{n} h_j^{(k,2l-2)} \right\},$$

and for $i = n, \ldots, 1$,

$$h_i^{(k,2l)} \leq \frac{1}{(n-1)} h_i^{(k,0)} \left\{ \sum_{j=1}^{i-1} h_j^{(k,2l-1)} + \sum_{j=i+1}^{n} h_j^{(k,2l)} \right\}.$$

Without loss of generality suppose that

$$h_i^{(0,0)} \leq h < 1 \quad (i = 1, \ldots, n).$$

Then it may be shown, as in the proof of Theorem 4.5.1, that

$$h_i^{(k,0)} \leq h^{(2r+1)^k} \quad (i = 1, \ldots, n)\,(\forall k \geq 0),$$

whence

$$O_R(\mathrm{PRSS1}, x_i^*) \geq 2r + 1 \quad (i = 1, \ldots, n). \square$$

4.4 The Interval Total-step and Single-step Procedures IT1 and IS1 for Simultaneously Bounding Real Polynomial Zeros

Let $p : R^1 \to R^1$ be a polynomial of degree n defined by

$$p(x) = \sum_{i=0}^{n} a_i x^i, \tag{4.4.1}$$

where $a_i \in R$ $(i = 0, \ldots, n)$. Suppose that p has n distinct zeros $x_i^* \in R$ $(i = 1, \ldots, n)$, and

that $\underline{x}_i^{(0)} \in I(R)$ $(i = 1, \ldots, n)$ are such that

$$x_i^* \in \underline{x}_i^{(0)} \quad (i = 1, \ldots, n), \tag{4.4.2}$$

and

$$\underline{x}_i^{(0)} \cap \underline{x}_j^{(0)} = \emptyset \quad (i, j = 1, \ldots, n; i \neq j). \tag{4.4.3}$$

As in §4.2, it is assumed henceforth that $a_n = 1$, so that

$$p(x) = \prod_{j=1}^{n} (x - x_j^*). \tag{4.4.4}$$

By 4.4.4, for $i = 1, \ldots, n$ $(\forall x \neq x_j^*$ $(j = 1, \ldots, n))$

$$x_i^* = x - \frac{p(x)}{\prod_{j \neq i} (x - x_j^*)}. \tag{4.4.5}$$

If

$$x_i^{(0)} = m(\underline{x}_i^{(0)}) \quad (i = 1, \ldots, n), \tag{4.4.6}$$

then by 4.4.2, 4.4.3

$$x_i^{(0)} \neq x_j^* \quad (i, j = 1, \ldots, n; j \neq i). \tag{4.4.7}$$

So by 4.4.5,

$$x_i^* = x_i^{(0)} - \frac{p(x_i^{(0)})}{\prod_{j \neq i} (x_i^{(0)} - x_j^*)} \quad (i = 1, \ldots, n). \tag{4.4.8}$$

Furthermore, by 4.4.3, 4.4.6, $x_i^{(0)} \notin \underline{x}_j^{(0)}$ $(i, j = 1, \ldots, n; j \neq i)$ whence

$$0 \notin \prod_{j \neq i} (x_i^{(0)} - \underline{x}_j^{(0)}) \quad (i = 1, \ldots, n). \tag{4.4.9}$$

So by 4.4.2, 4.4.8, and the inclusion monotonicity of real interval arithmetic,

$$x_i^* \in \underline{x}_i^{(1)} = \left\{ x_i^{(0)} - \frac{p(x_i^{(0)})}{\prod_{j \neq i} (x_i^{(0)} - \underline{x}_j^{(0)})} \right\} \cap \underline{x}_i^{(0)} \quad (i = 1, \ldots, n). \tag{4.4.10}$$

This gives rise to the total-step procedure IT1 [AleH--83] defined by

$$x_i^{(k)} = m(\underline{x}_i^{(k)}) \quad (i = 1, \ldots, n), \tag{4.4.11a}$$

$$\underline{x}_i^{(k+1)} = \left\{ x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j \neq i} (x_i^{(k)} - \underline{x}_j^{(k)})} \right\} \cap \underline{x}_i^{(k)} \quad (i = 1, \ldots, n)\ (k \geq 0), \tag{4.4.11b}$$

which may be regarded as an interval version of the procedure PT1. The following theorem is proved in [AleH--83].

Theorem 4.4.1

If (1) 4.4.2 and 4.4.3 hold; (2) the sequences $\left(\underline{x}_i^{(k)}\right)$ $(i = 1, \ldots, n)$ are generated from 4.4.11, then $(\forall k \geq 0)\ x_i^* \in \underline{x}_i^{(k+1)} \subseteq \underline{x}_i^{(k)}$ $(i = 1, \ldots, n)$. If also (3) $0 \notin \underline{d}_i$ where $\underline{d}_i \in I(R)$ is such that $p'(x) \in \underline{d}_i$ $(\forall x \in \underline{x}_i^{(0)})$ $(i = 1, \ldots, n)$, then $\underline{x}_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$ and $(\forall k \geq 0)\ (i = 1, \ldots, n)$

$$w(\underline{x}_i^{(k+1)}) \leq \tfrac{1}{2}\left(1 - \frac{d_{iI}}{d_{iS}}\right) w(\underline{x}_i^{(k)}). \qquad 4.4.12$$

Furthermore, for $i = 1, \ldots, n$, $O_R(\mathrm{IT1}, x_i^*) \geq 2$. □

The single-step procedure IS1 [AleH--83] is an interval version of the procedure PS1 and consists of generating the sequences $\left(\underline{x}_i^{(k)}\right)$ $(i = 1, \ldots, n)$ from

$$x_i^{(k)} = m(\underline{x}_i^{(k)}) \quad (i = 1, \ldots, n), \qquad 4.4.13a$$

$$\underline{x}_i^{(k+1)} = \left\{ x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k+1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k)})} \right\} \cap \underline{x}_i^{(k)} \qquad 4.4.13b$$

$$(i = 1, \ldots, n)\,(k \geq 0).$$

The following theorem is proved in [AleH--83].

Theorem 4.4.2

If (1) 4.4.2 and 4.4.3 hold; (2) the sequences $\left(\underline{x}_i^{(k)}\right)$ $(i = 1, \ldots, n)$ are generated from 4.4.13, then $(\forall k \geq 0)$ $x_i^* \in \underline{x}_i^{(k+1)} \subseteq \underline{x}_i^{(k)}$ $(i = 1, \ldots, n)$. If also (3) $0 \notin \underline{d}_i$ where $\underline{d}_i \in I(R)$ is such that $p'(x) \in \underline{d}_i$ $(\forall x \in \underline{x}_i^{(0)})$ $(i = 1, \ldots, n)$, then $\underline{x}_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$ and 4.4.12 holds. Furthermore, for $i = 1, \ldots, n$, $O_R(\mathrm{IS1}, x_i^*) \geq 1 + \sigma$ where $\sigma \in (1, 2)$ is the greatest positive zero of $t^n - t - 1$. □

4.5 The Repeated Symmetric Single-step Procedure IRSS1 for Simultaneously Bounding Real Polynomial Zeros

A natural extension of the procedure IS1 is the repeated symmetric single-step procedure IRSS1 which is based on the symmetric single-step idea [Ait---50],[Ale---77], and may be regarded as an interval version of the procedure PRSS1. The procedure IRSS1 consists of generating the sequences $\left(\underline{x}_i^{(k)}\right)$ $(i = 1, \ldots, n)$ from

$$\underline{x}_i^{(k,0)} = \underline{x}_i^{(k)} \quad (i = 1, \ldots, n), \tag{4.5.1a}$$

$$x_i^{(k)} = m(\underline{x}_i^{(k)}) \quad (i = 1, \ldots, n), \tag{4.5.1b}$$

$$p_i^{(k)} = p(x_i^{(k)}) \quad (i = 1, \ldots, n), \tag{4.5.1c}$$

$$\underline{x}_i^{(k,2l-1)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l-2)})} \right\} \cap \underline{x}_i^{(k,2l-2)} \tag{4.5.1d}$$

$$(i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l)})} \right\} \cap \underline{x}_i^{(k,2l-1)} \tag{4.5.1e}$$

$$(i = n, \ldots, 1),$$

$$\left(l = 1, \ldots, m^{(k)}\right),$$

$$\underline{x}_i^{(k+1)} = \underline{x}_i^{(k,2m^{(k)})} \quad (i = 1, \ldots, n)\,(k \geq 0), \tag{4.5.1f}$$

where $\left(m^{(k)}\right)$ is a sequence of positive integers. If $m^{(k)} = 1$ $(\forall k \geq 0)$ then the resulting procedure ISS1 may be regarded as an interval version of the symmetric single-step procedure PSS1. The procedure IRSS1 appears to be new.

Theorem 4.5.1

If (1) 4.4.2 and 4.4.3 hold; (2) the sequences $\left(\underline{x}_i^{(k)}\right)$ $(i = 1, \ldots, n)$ are generated from 4.5.1, then $(\forall k \geq 0)$ $x_i^* \in \underline{x}_i^{(k+1)} \subseteq \underline{x}_i^{(k)}$ $(i = 1, \ldots, n)$. If also (3) $0 \notin \underline{d}_i$ where $\underline{d}_i \in I(R)$ is such that $p'(x) \in \underline{d}_i$ $(\forall x \in \underline{x}_i^{(0)})$ $(i = 1, \ldots, n)$, then $\underline{x}_i^{(k)} \to x_i^*$ $(k \to \infty)$ $(i = 1, \ldots, n)$ and 4.4.12 holds. Furthermore, if $m^{(k)} = m$ $(\forall k \geq 0)$ then for $(i = 1, \ldots, n)$, $O_R(\text{IRSS1}, x_i^*) \geq (2m + 1)$.

Proof

The proof that $x_i^* \in \underline{x}_i^{(k+1)} \subseteq \underline{x}_i^{(k)}$ $(i = 1, \ldots, n)$ $(\forall k \geq 0)$ and that 4.4.12 holds is almost identical with the corresponding proofs in Theorem 4.4.1 and Theorem 4.4.2, and is therefore omitted. It remains to prove that $O_R(\text{IRSS1}, x_i^*) \geq (2m + 1)$ $(i = 1, \ldots, n)$ if $m^{(k)} = m$ $(\forall k \geq 0)$.

As in the proof of Theorem 4.4.2 [AleH--83] it may be shown that $\exists \alpha > 0$ such that $(\forall k \geq 0)$ $(l = 1, \ldots, m)$,

$$w_i^{(k,2l-1)} \leq \beta w_i^{(k,0)} \left\{ \sum_{j=1}^{i-1} w_j^{(k,2l-1)} + \sum_{j=i+1}^{n} w_j^{(k,2l-2)} \right\} \quad (i = 1, \ldots, n), \qquad 4.5.2$$

and

$$w_i^{(k,2l)} \leq \beta w_i^{(k,0)} \left\{ \sum_{j=1}^{i-1} w_j^{(k,2l-1)} + \sum_{j=i+1}^{n} w_j^{(k,2l)} \right\} \quad (i = n, \ldots, 1), \qquad 4.5.3$$

where

$$w_i^{(k,r)} = (n-1)\alpha w(\underline{x}_i^{(k,r)}) \quad (i = 1, \ldots, n)\,(r = 0, \ldots, 2m), \tag{4.5.4}$$

and

$$\beta = 1/(n-1). \tag{4.5.5}$$

For $l = 1, \ldots, m$ let

$$u_i^{(1,2l-1)} = \begin{cases} 2l & (i = 1, \ldots, n-1) \\ \\ 2l+1 & (i = n) \end{cases}, \tag{4.5.6}$$

and

$$u_i^{(1,2l)} = \begin{cases} 2l+2 & (i = 1) \\ \\ 2l+1 & (i = 2, \ldots, n) \end{cases}, \tag{4.5.7}$$

and for $r = 1, \ldots, 2m$ let

$$u_i^{(k+1,r)} = \begin{cases} (2m+1)u_i^{(k,r)} + 1 & (i = 1) \\ \\ (2m+1)u_i^{(k,r)} & (i = 2, \ldots, n) \end{cases}. \tag{4.5.8}$$

Then for $l = 1, \ldots, m$ ($\forall k \geq 0$), by 4.5.6–4.5.8,

$$u_i^{(k,2l-1)} = \begin{cases} \{(4ml+1)/(2m)\}(2m+1)^{k-1} - 1/(2m) & (i=1) \\ 2l(2m+1)^{k-1} & (i=2,\ldots,n-1)\,, \\ (2l+1)(2m+1)^{k-1} & (i=n) \end{cases} \quad 4.5.9$$

and

$$u_i^{(k,2l)} = \begin{cases} \{(4ml+4m+1)/(2m)\}(2m+1)^{k-1} - 1/(2m) & (i=1) \\ (2l+1)(2m+1)^{k-1} & (i=2,\ldots,n) \end{cases} . \quad 4.5.10$$

Suppose, without loss of generality, that

$$w_i^{(0,0)} \leq h < 1 \quad (i=1,\ldots,n). \quad 4.5.11$$

Then by a lengthy inductive argument it follows from 4.5.2–4.5.11 that for $l = 1,\ldots,m$, $i = 1,\ldots,n$, $k \geq 0$,

$$w_i^{(k,2l-1)} \leq h^{u_i^{(k+1,2l-1)}},$$

and

$$w_i^{(k,2l)} \leq h^{u_i^{(k+1,2l)}},$$

whence, by 4.5.10 with $l = m$ and 4.5.1f, $(\forall k \geq 0)$

$$w_i^{(k+1)} \leq h^{(2m+1)^{(k+1)}} \quad (i = 1, \ldots, n).$$

So ($\forall k \geq 0$), by 4.5.3–4.5.11,

$$w(\underline{x}_i^{(k)}) \leq (\beta/\alpha)h^{(2m+1)^k} \quad (i = 1, \ldots, n). \qquad 4.5.12$$

Let

$$w^{(k)} = \max_{1 \leq i \leq n} \{ w(\underline{x}_i^{(k)}) \}.$$

Then by 4.5.12,

$$w^{(k)} \leq (\beta/\alpha)h^{(2m+1)^k} \quad (\forall k \geq 0).$$

So

$$R_{2m+1}(w^{(k)}) = \limsup_{k \to \infty} \{ (w^{(k)})^{1/(2m+1)^k} \}$$

$$\leq \limsup_{k \to \infty} \{ (\beta/\alpha)^{1/(2m+1)^k} h \}$$

$$= h$$

$$< 1.$$

Therefore [AleH--83] [OrtR--70],

$$O_R(\text{IRSS1}, x_i^*) \geq (2m + 1) \quad (i = 1, \ldots, n). \square$$

4.6 The Determination of $m^{(k)}$ in IRSS1

An iteration of IRSS1 consists of an outer iteration in which $\underline{x}^{(k,2)} \in I(R^n)$ is computed from $\underline{x}^{(k)}$ using 4.5.1 with $l = 1$ and $m^{(k)} - 1$ inner iterations in which $\underline{x}^{(k,2l)}$ is computed from $\underline{x}^{(k,2l-2)}$ using 4.5.1*d* and 4.5.1*e* with $l = 2, \ldots, m^{(k)}$.

If T_O and T_I are the cpu times which are required for one outer iteration and one inner iteration respectively, then

$$T_O = t_1 + t_2, \tag{4.6.1}$$

and

$$T_I = 2t_2, \tag{4.6.2}$$

where t_1 and t_2 are the cpu times which are required to compute $\underline{x}^{(k,1)}$ from $\underline{x}^{(k)}$ using 4.5.1*a*–4.5.1*d* and to compute $\underline{x}^{(k,2)}$ from $\underline{x}^{(k,1)}$ using 4.5.1*e* respectively: $t_1 > t_2$ because t_1 includes the cpu time which is required to compute $p(x_i^{(k)})$ $(i = 1, \ldots, n)$. In practice it is sufficient to measure t_1 and t_2 once only (when $k = 0$).

Suppose that, for some $k \geq 0$, and for some $l \geq 1$, $\underline{x}^{(k,1)}, \ldots, \underline{x}^{(k,2l)}$ have been computed. It is necessary to decide whether to set $m^{(k)} = l$ and then compute an outer iterate, or to compute an inner iterate. Computational experience indicates that indices ρ_O and ρ_I of efficiency for the next outer iteration and for the next inner iteration respectively are given by

$$\rho_O = -ln\left(||w(\underline{x}^{(k+1,2)})||/||w(\underline{x}^{(k,2l)})||\right)/T_O, \tag{4.6.3}$$

and

$$\rho_I = -ln\left(||w(\underline{x}^{(k,2l+2)})||/||w(\underline{x}^{(k,2l)})||\right)/T_I, \qquad 4.6.4$$

where $\underline{x}^{(k+1,2)}$ would be computed from 4.5.1*a*–4.5.1*e* with $\underline{x}^{(k+1)} = \underline{x}^{(k,2l)}$.

In order to use 4.6.3 and 4.6.4 it is necessary to estimate

$$||w(\underline{x}^{(k+1,2)})||/||w(\underline{x}^{(k,2l)})||,$$

and

$$||w(\underline{x}^{(k,2l+2)})||/||w(\underline{x}^{(k,2l)})||.$$

For $k \geq 0$ and $r \geq 0$ let $w^{(k,r)} = ||w(\underline{x}^{(k,r)})||$. Then 4.5.2 and 4.5.3 suggest that if, for $k \geq 0$ and $r \geq 1$ (see Figure 4.6.1)

$$Q^{(k)} = w^{(k,1)}/(w^{(k,0)})^2, \qquad 4.6.5$$

and

$$L^{(k,r)} = w^{(k,r+1)}/(w^{(k,0)}w^{(k,r)}), \qquad 4.6.6$$

then $Q^{(k+1)} \approx Q^{(k)}$ and $L^{(k,r+1)} \approx L^{(k,r)}$ where $\approx$ denotes approximate equality. Computational experience indicates that $Q^{(k)}$ and $L^{(k,r)}$ remain reasonably constant from one iteration to the next. Therefore if l is as in 4.6.3 and 4.6.4 then by 4.6.5 and 4.6.6

$$w^{(k+1,1)} \approx Q^{(k)}(w^{(k,2l)})^2,$$

and

$$w^{(k+1,2)} \approx L^{(k,2l-1)}w^{(k,2l)}w^{(k+1,1)},$$

whence

$$w^{(k+1,2)}/w^{(k,2l)} \approx M^{(k)}(w^{(k,2l)})^2, \qquad 4.6.7$$

where

$$M^{(k)} = w^{(k,2)}/(w^{(k,0)})^3, \qquad 4.6.8$$

and

$$w^{(k,2l+2)}/w^{(k,2l)} \approx N^{(k,l)}w^{(k,2l)}, \qquad 4.6.9$$

where

$$N^{(k,l)} = w^{(k,2l)}/(w^{(k,2l-1)})^2. \qquad 4.6.10$$

From 4.6.3, 4.6.4, and 4.6.7–4.6.10,

$$\rho_O \approx -ln\left(M^{(k)}(w^{(k,2l)})^2\right)/T_O, \qquad 4.6.11$$

and

$$\rho_I \approx -ln\left(N^{(k,l)}w^{(k,2l)}\right)/T_I. \qquad 4.6.12$$

If $\rho_O > \rho_I$ then $\underline{x}^{(k+1,1)}$, $\underline{x}^{(k+1,2)}$, and $M^{(k+1)}$ are computed from 4.5.1*a*–4.5.1*d*, 4.5.1*e*, and 4.6.8 respectively.

If $\rho_O \leq \rho_I$ then $\underline{x}^{(k,2l+1)}$, $\underline{x}^{(k,2l+2)}$, and $N^{(k,l+1)}$ are computed from 4.5.1*d*, 4.5.1*e*, and 4.6.10 respectively.

If the predicted value of $w^{(k+1,2)}$ is too small to be attainable by the computer then an inner iteration is performed because the assumption that $M^{(k+1)} \approx M^{(k)}$ is no longer valid due to rounding error. A similar phenomenon has been observed by Alefeld and Platzöder [AleP--83]. Therefore an inner iteration is always performed if $\bar{w}^{(k+1,2)} \leq \varepsilon_M$ where $\bar{w}^{(k+1,2)}$ is the predicted value of $w^{(k+1,2)}$ and ε_M is machine dependent.

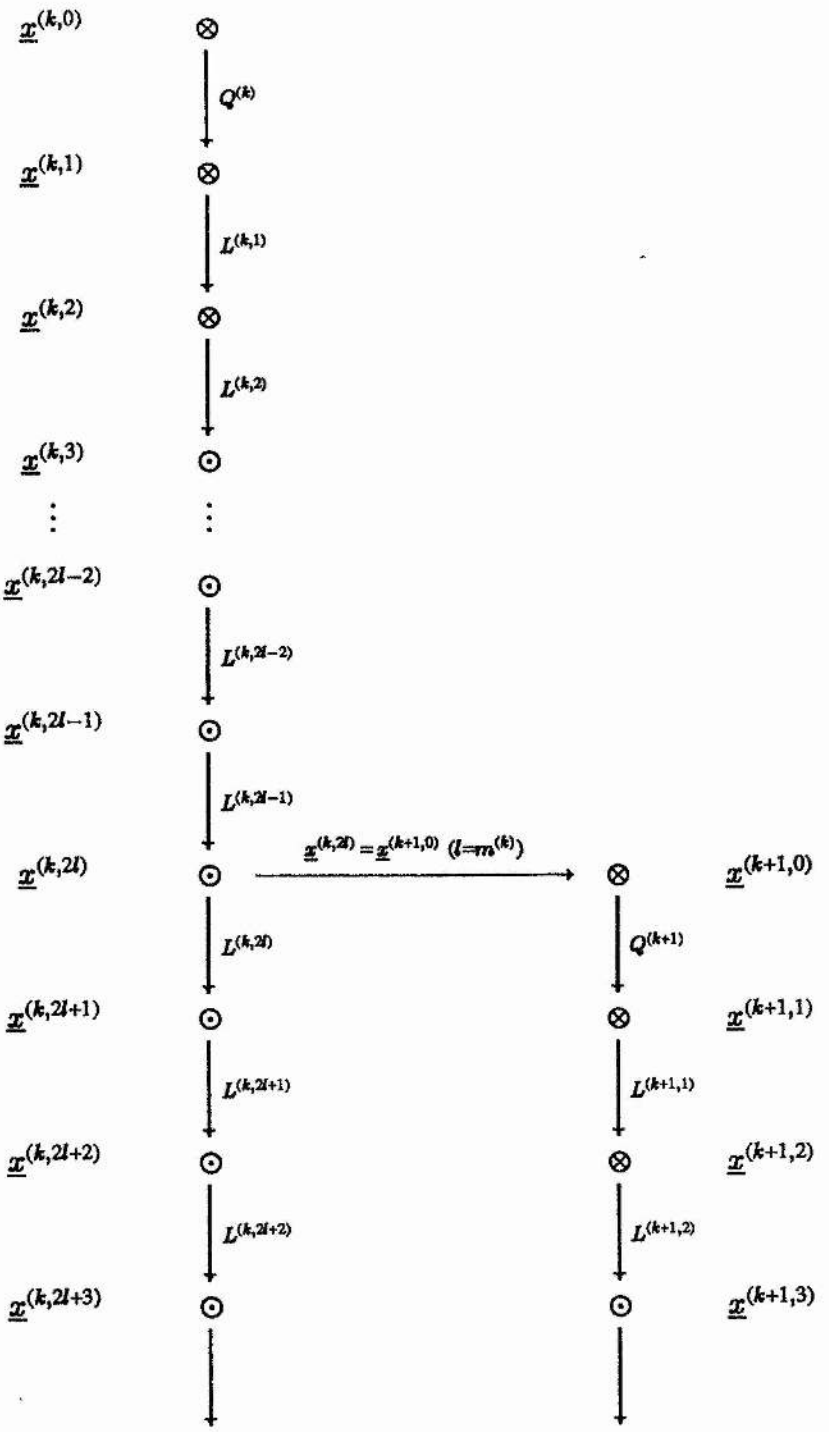


Figure 4.6.1: Outer iteration (⊗) and inner iteration (⊙).

4.7 Numerical Results

In order to construct an efficient algorithm for IRSS1 the following observations must be taken into account.

(1) For $l = 1, \ldots, m^{(k)}$ the products

$$\prod_{j=1}^{i-1} (x_i^{(k)} - \underline{x}_j^{(k,2l-1)}) \quad (i = 2, \ldots, n)$$

which occur in 4.5.1*d* should be saved for re-used in 4.5.1*e*.

(2) For $l = 1, \ldots, m^{(k)}$ the products

$$\prod_{j=i+1}^{n} (x_i^{(k)} - \underline{x}_j^{(k,2l)}) \quad (i = 1, \ldots, n-1)$$

which occur in 4.5.1*e* should be saved for re-use in 4.5.1*d* (*i.e.* for the next l).

(3) For $k \geq 0$, $l = 1, \ldots, m^{(k)}$, $\underline{x}_n^{(k,2l)} = \underline{x}_n^{(k,2l-1)}$ and for $k \geq 1$, $l = 1, \ldots, m^{(k)}$, $\underline{x}_1^{(k,2l-1)} = \underline{x}_1^{(k,2l-2)}$, so in the first iteration, $\underline{x}_n^{(0,2l)}$ need not be computed, and in all subsequent iterations, neither $\underline{x}_n^{(k,2l)}$ nor $\underline{x}_1^{(k,2l-1)}$ need be computed.

(4) A strategy such as that which is described in §4.6 for determining $m^{(k)}$ should be used.

Algorithms corresponding to the methods IT1, IS1, ISS1 and IRSS1 have been implemented in Triplex S-algol [BaiCM-82] [McbWs-83] on a VAX 11-785 computer. Numerical

results for 5 examples (see Appendix A) are given in this section. For each example the stopping criterion is

$$w^{(k,0)} \leq 10^{-10},$$

and $\varepsilon_M = 10^{-17}$. Table 4.7.1a contains the cpu times corresponding to Examples 4.1–4.5 and algorithms IT1, IS1, ISS1, where, in IRSS1, the strategy for determining $m^{(k)}$ which is described in §4.6 is used. The algorithm ISS1 is obtained by setting $m^{(k)} = 1$ $(\forall k \geq 0)$ in IRSS1. Table 4.7.2a contains the cpu times corresponding to Examples 4.1–4.5 for the algorithm IRSS1 with $m^{(k)} = 1$ $(\forall k \geq 0)$ for $m = 1, \ldots, 6$. Tables 4.7.3a contains the cpu times corresponding to Example 4.5 with $n = 2, 4, 6, \ldots, 14$ for the algorithm IRSS1 with $m^{(k)} = m$ $(\forall k \geq 0)$ for $m = 1, \ldots, 6$, and for the algorithm IRSS1 with $m^{(k)}$ determined by using the strategy which is described in §4.6 (under *). Tables 4.1b–4.3b contain the number of iterations corresponding to the cpu times in Tables 4.1a–4.3a.

Table 4.7.1 illustrates the fact that IRSS1 with automatic determination of $m^{(k)}$ requires less cpu time and number of iterations than does IT1, IS1, and ISS1.

Table 4.7.2a illustrates the fact that the cpu time required by IRSS1, with $m^{(k)} = m$ $(\forall k \geq 0)$ where $m \geq 1$ is selected by user, varies significantly with m and that often $m = 2$ is a near optimal choice.

Tables 4.7.3 illustrates the facts that, at least for the examples which have been tried, IRSS1 with automatic determination of $m^{(k)}$ usually requires less cpu time than does IRSS1 with $m^{(k)} = m$ $(\forall k \geq 0)$ even m is chosen optimally, and that the saving in cpu time which is obtained by using automatic determination of $m^{(k)}$ increases with the degree n of the polynomial.

The algorithm corresponding to IRSS1 with automatic determination of $m^{(k)}$ has been implemented in rectangular interval arithmetic [RokL--71][RokL--75] and preliminary results indicate that its behaviour is similar to that of the implementation in which real interval arithmetic is used.

Example	n	IT1	IS1	ISS1	IRSS1
4.1	9	3.67	3.06	2.92	2.43
4.2	5	1.23	1.15	1.14	0.97
4.3	9	4.28	3.80	3.65	3.50
4.4	9	4.41	3.71	3.71	3.48
4.5	14	9.76	8.09	6.27	5.46

Table 4.7.1a: Cpu times.

Example	n	IT1	IS1	ISS1	IRSS1
4.1	9	5	4	3	2
4.2	5	4	4	3	2
4.3	9	6	5	4	3
4.4	9	6	5	4	3
4.5	14	6	5	3	2

Table 4.7.1b: Number of iterations.

m Example	1	2	3	4	5	6
4.1	2.92	2.56	3.53	4.20	2.45	2.92
4.2	1.14	1.04	1.31	1.64	0.92	1.12
4.3	3.65	3.95	3.42	4.20	5.14	5.88
4.4	3.71	3.95	3.48	4.36	4.47	5.76
4.5	6.27	5.84	7.52	9.24	11.07	12.71

Table 4.7.2a: Cpu times.

m Example	1	2	3	4	5	6
4.1	3	2	2	2	1	1
4.2	3	2	2	2	1	1
4.3	4	3	2	2	2	2
4.4	4	3	2	2	2	2
4.5	3	2	2	2	2	2

Table 4.7.2b: Number of iterations.

m / n	1	2	3	4	5	6	*
2	0.20	0.24	0.29	0.38	0.22	0.24	0.30
4	0.77	0.71	0.98	1.17	1.38	1.60	0.77
6	1.40	1.42	1.77	2.16	2.55	2.84	1.37
8	2.32	2.20	2.87	3.45	4.26	4.71	2.14
10	3.59	3.20	4.16	5.14	6.08	6.91	1.89
12	4.73	4.52	5.97	7.07	8.21	9.87	4.26
14	6.27	5.84	7.52	9.24	11.07	12.71	5.46

Table 4.7.3a: Cpu times.

m / n	1	2	3	4	5	6	*
2	2	2	2	2	1	1	2
4	3	2	2	2	2	2	2
6	3	2	2	2	2	2	2
8	3	2	2	2	2	2	2
10	3	2	2	2	2	2	2
12	3	2	2	2	2	2	2
14	3	2	2	2	2	2	2

Table 4.7.3b: Number of iterations.

CHAPTER 5

Interval Versions of some Procedures for the Simultaneous Estimation of Simple Polynomial Zeros

5.1 Introduction

Let the polynomial $p : C \rightarrow C$ be defined by

$$p(z) = \sum_{i=0}^{n} a_i z^i \qquad 5.1.1$$

in which $a_n = 1$ and suppose that p has n simple zeros $z_i^* \in C$ $(i = 1, \ldots, n)$. Unfortunately, interval arithmetic is not at present available in most high level programming languages, so that interval arithmetic operations are usually performed through procedure invocations, leading to larger cpu times than for real or complex point arithmetic. Therefore it is desirable that as much of the computation as possible should be done in point arithmetic without losing the computational rigour of the bounds on the polynomial zeros.

It is shown in this chapter that an idea due to Neumaier [Neu---84] [Neu---85] can be used to obtain interval versions of point iterative procedures for the simultaneous inclusion of simple polynomial zeros. These interval procedures have several advantages over existing point and interval procedures for simple polynomial zeros: (a) simple computationally verifiable existence, uniqueness and convergence tests exist; (b) the convergence of the corresponding point iterative procedure is forced; (c) the asymptotic convergence rate of the interval sequences containing the zeros is equal to that of the point sequences; (d) the cpu time required to satisfy a given stopping criterion is much less than that which is required by corresponding existing interval procedures.

The remainder of this chapter is organized as follows. Section 5.2 contains preliminary material which is required in Section 5.3. In Section 5.3, some recent ideas due to Neumaier [Neu---84],[Neu---85] are used to obtain an interval version IP of a point procedure P for the estimation of a zero of an analytic function $f : C \rightarrow C$. Existence, uniqueness, and convergence results similar to those which have been described in [Neu---85] are given. In Section 5.4 the procedure IP is used to construct interval versions IPT1, IPS1, IPSS1, and IPRSS1 of the point procedures PT1, PS1, PSS1, and PRSS1 which are described in Section 4.3. Section 5.5 contains numerical results which are obtained by using the procedures IPT1, IPS1, IPSS1, IPRSS1, and the interval iterative procedures IT1, IS1, ISS1, and IRSS1 (§4.5) to bound the zeros of 6 polynomials (see Appendix A). Finally, ideas for future work are presented in Section 5.6.

5.2 Preliminaries

The following result, which is not given in [AleH--83] appears to be well known but does not seem to be readily available in the literature.

Lemma 5.2.1

Let $f : C \rightarrow C$ be a given function, let $S \subseteq C$ be an open convex set, and let $\hat{\underline{z}} \in I_R(S)$ be given. If (1) f is analytic in S; (2) $\exists z^* \in \hat{\underline{z}}$ such that $f(z^*) = 0$; (3) $\underline{f'} : I_R(S) \rightarrow I_R(C)$ is a continuous inclusion monotonic interval extension of the derivative $f' : S \rightarrow C$ of f, then $(\forall z \in \hat{\underline{z}})$

$$|f(z)|_R \leq |\underline{f'}(\hat{\underline{z}})|_R |z - z^*|_R.$$

Proof

Let $z \in \hat{\underline{z}}$ be given and let $z = x + iy$, $z^* = x^* + iy^*$. If $f(z) = u(x, y) + iv(x, y)$,

then by the Cauchy-Riemann relations [Chu---60] [Hen---74] and the mean value theorem, $\exists \theta_j \in [0, 1]$ $(j = 1, 2)$ such that

$$u(x, y) = \partial_1 u(\xi_1, \eta_1)(x - x^*) - \partial_1 v(\xi_1, \eta_1)(y - y^*),$$

and

$$v(x, y) = \partial_1 v(\xi_2, \eta_2)(x - x^*) + \partial_1 u(\xi_2, \eta_2)(y - y^*),$$

where

$$\partial_1 u(x, y) = \frac{\partial}{\partial x} u(x, y),$$

$$\partial_1 v(x, y) = \frac{\partial}{\partial x} v(x, y),$$

and for $j = 1, 2$,

$$\xi_j = x^* + \theta_j(x - x^*),$$

and

$$\eta_j = y^* + \theta_j(y - y^*).$$

So by Hypothesis (3),

$$|f(z)|_R = |u(x, y)| + |v(x, y)|$$

$$\leq \left\{ |\partial_1 u(\xi_1, \eta_1)| + |\partial_1 v(\xi_2, \eta_2)| \right\} |x - x^*| +$$

$$\left\{|\partial_1 u(\xi_2,\eta_2)| + |\partial_1 v(\xi_1,\eta_1)|\right\}|y - y^*|$$

$$\leq \left\{|(\underline{f}'(\hat{z}))_R| + |(\underline{f}'(\hat{z}))_I|\right\}(|x - x^*| + |y - y^*|)$$

$$= |\underline{f}'(\hat{z})|_R\,|z - z^*|_R. \quad \square$$

5.3 Interval Inclusions of the Zeros of Analytic Functions

Recently Neumaier [Neu---84],[Neu---85] has shown how to use locally convergent point iterative procedures for the estimation of zeros of appropriate functions $f : R^n \to R^n$ ($n \geq 1$) to construct interval procedures for the inclusion of the zeros which force global convergence of the locally convergent point procedures without sacrificing the asymptotic convergence rate of the point procedures. It is shown in this section that Neumaier's arguments may be used to construct interval procedures in which rectangular complex interval arithmetic is used to bound simple zeros of analytic functions $f : C \to C$.

Theorem 5.3.1

Let $f : C \to C$ be a given function which is analytic in an open convex set $S \subseteq C$. Let $\hat{z} \in I_R(S)$ be given, and let $\underline{\hat{D}} \in I_R(C)$ be such that $0 \notin \underline{\hat{D}}$ and if $f(z) = u(x,y) + iv(x,y)$ where $z = x + iy \in S$ then

$$\partial_1 u(x',y') + i\partial_1 v(x'',y'') \in \underline{\hat{D}} \quad (\forall x', x'' \in \hat{z}_R)(\forall y', y'' \in \hat{z}_I)$$

in which $\underline{\hat{z}} = \underline{\hat{z}}_R + i\underline{\hat{z}}_I$. Then

(a) $(z^* \in \underline{\hat{z}} \wedge f(z^*) = 0 \wedge \hat{z} \in \underline{\hat{z}}) \Rightarrow (z^* \in \hat{z} - f(\hat{z})/\underline{\hat{D}})$;

(b) $(z^*, z^{**} \in \underline{\hat{z}} \wedge f(z^*) = f(z^{**}) = 0) \Rightarrow (z^{**} = z^*)$;

(c) $(\hat{z} \in int(\underline{\hat{z}}) \wedge \hat{z} - f(\hat{z})/\underline{\hat{D}} \subseteq \underline{\hat{z}}) \Rightarrow (\exists z^* \in \underline{\hat{z}}, f(z^*) = 0)$.

Proof

(a) Let $z', z'' \in \hat{\underline{z}}$ be fixed. Then $z'' + t(z' - z'') \in \hat{\underline{z}}$ $(\forall t \in [0, 1])$ because $\hat{\underline{z}}$ is convex. Let $\psi : [0, 1] \to C$ be defined by

$$\psi(t) = f'(z'' + t(z' - z'')).$$

Then $\psi(t) \in \hat{\underline{D}}$ $(\forall t \in [0, 1])$. If

$$\psi(t) = U(t) + iV(t),$$

then by definition of Riemann integration in C,

$$\int_a^b \psi(t)\,dt = \int_a^b U(t)\,dt + i\int_a^b V(t)\,dt \quad (0 \leq a \leq b \leq 1).$$

Now since f is analytic in S, f' is continuous so U and V are continuous. So by the mean value theorem of integration $\exists \xi, \eta \in [a, b]$ such that

$$\int_a^b U(t)\,dt = U(\xi)(b - a)$$

and

$$\int_a^b V(t)\,dt = V(\eta)(b - a).$$

So if $a = 0, b = 1$, then

$$\int_0^1 \psi(t)\,dt = \int_0^1 U(t)\,dt + i\int_0^1 V(t)\,dt$$

$$= U(\xi) + iV(\eta).$$

Now by definition of $\underline{\hat{D}}$, $(\forall x', x'' \in \underline{\hat{z}}_R)\ (\forall y', y'' \in \underline{\hat{z}}_I)$,

$$\partial_1 u(x', y') + i\partial_1 v(x'', y'') \in \underline{\hat{D}}.$$

Also

$$\psi(t) = f'(z'' + t(z' - z''))$$

$$= U(z'' + t(z' - z'')) + iV(z'' + t(z' - z'')).$$

So

$$\int_0^1 \psi(t)\,dt = U(z'' + \xi(z' - z'')) + iV(z'' + \eta(z' - z'')).$$

If

$$f(z) = u(x, y) + iv(x, y),$$

then

$$f'(z) = \partial_1 u(x, y) + i\partial_1 v(x, y).$$

So

$$\psi(t) = f'(z'' + t(z' - z''))$$

$$= \partial_1 u(x'' + t(x' - x''), y'' + t(y' - y''))$$

$$+ i\partial_1 v(x'' + t(x' - x''), y'' + t(y' - y'')).$$

Therefore

$$U(t) = \partial_1 u(x'' + t(x' - x''), y'' + t(y' - y''))$$

and

$$V(t) = \partial_1 v(x'' + t(x' - x''), y'' + t(y' - y'')).$$

So by definition of $\underline{\hat{D}}$,

$$U(\xi) + iV(\eta) \in \underline{\hat{D}} \quad (\forall \xi, \eta \in [0, 1]),$$

whence

$$\int_0^1 \psi(t)\, dt = U(\xi) + iV(\eta) \in \underline{\hat{D}}.$$

Let

$$\hat{d} = \int_0^1 \psi(t)\, dt.$$

Then $\hat{d} \in \underline{\hat{D}}$. Let $g : [0, 1] \to S$ be continuous, and suppose that g' exists on $[0, 1]$. Then since f is analytic in S,

$$\frac{d}{dt} f(g(t)) = \frac{df}{dg} \cdot \frac{dg}{dt}$$

$$= f'(g(t)) \cdot g'(t).$$

Let

$$g(t) = z'' + t(z' - z'').$$

Then

$$g'(t) = z' - z''.$$

So if

$$\phi(t) = f(g(t))$$

$$= f(z'' + t(z' - z'')),$$

then

$$\phi'(t) = f'(g(t)) \cdot g'(t)$$

$$= f'(z'' + t(z' - z''))(z' - z'')$$

$$= \psi(t)(z' - z'').$$

So

$$\hat{d}(z' - z'') = \int_0^1 \psi(t)(z' - z'')\, dt$$

$$= \int_0^1 \phi'(t)\, dt$$

$$= \phi(1) - \phi(0)$$

$$= f(z') - f(z'').$$

Now

$$\hat{d}(z' - z'') = \phi(1) - \phi(0)$$

$$= f(z') - f(z'').$$

So

$$z'' = z' - \frac{1}{\hat{d}}(f(z') - f(z'')).$$

Suppose that $\tau \in C$ is such that

$$f(z'') = \tau f(z').$$

Then

$$z'' = z' - (1 - \tau)\frac{f(z')}{\hat{d}}$$

$$\in z' - (1 - \tau)\frac{f(z')}{\underline{\hat{D}}}.$$

Now $z', z'' \in \underline{\hat{z}}$ are arbitrary. Let $z'' = z^*$, $z' = \hat{z}$. Then $f(z^*) = 0$, and if $f(\hat{z}) \neq 0$ then

$$(f(z^*) = \tau f(\hat{z})) \Rightarrow (\tau = 0).$$

So

$$z^* \in \hat{z} - (1 - \tau)\frac{f(\hat{z})}{\underline{\hat{D}}}$$

$$= \hat{z} - \frac{f(\hat{z})}{\underline{\hat{D}}}.$$

This proves (a).

(b) Let $z^*, z^{**} \in \underline{\hat{z}}$ be such that $f(z^*) = f(z^{**}) = 0$. Then by (a)

$$z^* \in z^{**} - \frac{f(z^{**})}{\underline{\hat{D}}}$$

$$= z^{**}.$$

So $z^{**} = z^*$. This proves (b).

(c) As shown in the proof of (a), if $z', z'' \in \underline{\hat{z}}$ then

$$z'' = z' - \frac{1}{\hat{d}}(f(z') - f(z'')),$$

where

$$\hat{d} = \int_0^1 f'(z'' + t(z' - z''))\, dt.$$

So

$$f(z') - f(z'') = -\hat{d}(z'' - z').$$

Let $z' = z, z'' = \hat{z}$. Then

$$f(z) - f(\hat{z}) = -\hat{d}(\hat{z} - z).$$

Let

$$\chi(z) = z - \frac{f(z)}{\hat{d}}.$$

Then $(\forall z \in \underline{\hat{z}})$,

$$\chi(z) = z - \frac{f(\hat{z})}{\hat{d}} + \frac{(f(\hat{z}) - f(z))}{\hat{d}}$$

$$= z - \frac{f(\hat{z})}{\hat{d}} + \hat{z} - z$$

$$= \hat{z} - \frac{f(\hat{z})}{\hat{d}}$$

$$\in \hat{z} - \frac{f(\hat{z})}{\underline{\hat{D}}}.$$

So

$$\chi(z) \in \hat{z} - \frac{f(\hat{z})}{\underline{\hat{D}}} \quad (\forall z \in \hat{\underline{z}}).$$

Therefore if $\hat{z} - f(\hat{z})/\underline{\hat{D}} \subseteq \hat{\underline{z}}$ then $\chi(z) \in \hat{\underline{z}}$ $(\forall z \in \hat{\underline{z}})$. So by Brouwer's theorem, $\exists z^* \in \hat{\underline{z}}$ such that $f(z^*) = 0$. This proves (c). □

Theorem 5.3.2

Let $f, S, \hat{\underline{z}}$, and $\underline{\hat{D}}$ be as in Theorem 5.3.1. Let the sequence $\left(\underline{z}^{(k)}\right)$ be generated from

$$\underline{z}^{(k+1)} = \left\{z^{(k)} - \frac{f(z^{(k)})}{\underline{\hat{D}}}\right\} \cap \underline{z}^{(k)} \quad (k \geq 0) \qquad 5.3.1$$

with $\underline{z}^{(0)} = \hat{\underline{z}}$ and $z^{(k)} \in \underline{z}^{(k)}$ arbitrary $(\forall k \geq 0)$. Then

(a) $(z^* \in \hat{\underline{z}} \wedge f(z^*) = 0) \Rightarrow (z^* \in \underline{z}^{(k+1)} \subseteq \underline{z}^{(k)}\ (\forall k \geq 0))$;

(b) $(\exists k \geq 0, \underline{z}^{(k)} = \emptyset) \Rightarrow (\nexists z^* \in \hat{\underline{z}}, f(z^*) = 0)$;

(c) $(z^* \in \hat{\underline{z}} \wedge f(z^*) = 0) \Rightarrow (\underline{z}^{(k)} \to z^*\ (k \to \infty) \wedge z^{(k)} \to z^*\ (k \to \infty))$;

(d) $(\nexists z^* \in \hat{\underline{z}}, f(z^*) = 0) \Rightarrow (\exists k \geq 0, \underline{z}^{(k)} = \emptyset)$.

Proof

(a) By Theorem 5.3.1(a),

$$\left(z^* \in \underline{z}^{(0)}\right) \Rightarrow \left(z^* \in z^{(0)} - \frac{f(z^{(0)})}{\underline{\hat{D}}}\right).$$

So

$$z^* \in \underline{z}^{(1)} = \left\{z^{(0)} - \frac{f(z^{(0)})}{\underline{\hat{D}}}\right\} \cap \underline{z}^{(0)}.$$

Therefore, by induction on k

$$z^* \in \underline{z}^{(k+1)} \subseteq \underline{z}^{(k)} \quad (\forall k \geq 0).$$

This proves (a).

(b) Suppose $\exists z^* \in \hat{\underline{z}}$ such that $f(z^*) = 0$. Then by (a), $z^* \in \underline{z}^{(k)}$ $(\forall k \geq 0)$. So $\underline{z}^{(k)} \neq \emptyset$ $(\forall k \geq 0)$. So if $\underline{z}^{(k)} = \emptyset$ for some k, then $\nexists z^* \in \hat{\underline{z}}$ such that $f(z^*) = 0$. This proves (b).

(c) Now $z^{(k)} \in \underline{z}^{(k)} \subseteq \hat{\underline{z}}$ $(\forall k \geq 0)$. So $z^{(k)}$ lies in the compact set $\hat{\underline{z}}$ $(\forall k \geq 0)$. Therefore if $\underline{z}^{(k)} \neq \emptyset$ $(\forall k \geq 0)$, $\left(\underline{z}^{(k)}\right)$ contains a convergent subsequence $\left(z^{(k_i)}\right)$. Suppose that $z^{(k_i)} \to \tilde{z}$ $(i \to \infty)$. Then $(\forall i \geq 0)$ since

$$\left(z^{(k+1)} \in \underline{z}^{(k+1)} \subseteq \underline{z}^{(k)} \ (\forall k \geq 0)\right) \Rightarrow \left(\tilde{z} \in \underline{z}^{(k)} \ (\forall k \geq 0)\right),$$

it follows that

$$\tilde{z} \in \underline{z}^{(k_i+1)} \subseteq z^{(k_i)} - \frac{f(z^{(k_i)})}{\hat{\underline{D}}},$$

whence on proceeding to the limit (since f is continuous)

$$\tilde{z} \in \tilde{z} - \frac{f(\tilde{z})}{\hat{\underline{D}}},$$

so that

$$0 \in \frac{f(\tilde{z})}{\hat{\underline{D}}}.$$

But $0 \notin \underline{\hat{D}}$. So $f(\tilde{z}) = 0$. Now by continuity of f,

$$z^{(k_i)} - \frac{f(z^{(k_i)})}{\underline{\hat{D}}} \to \tilde{z} - \frac{f(\tilde{z})}{\underline{\hat{D}}} \quad (i \to \infty).$$

So because $f(\tilde{z}) = 0$, it follows that

$$z^{(k_i)} - \frac{f(z^{(k_i)})}{\underline{\hat{D}}} \to \tilde{z} \quad (i \to \infty).$$

So

$$\underline{z}^{(k_i+1)} = \left\{ z^{(k_i)} - \frac{f(z^{(k_i)})}{\underline{\hat{D}}} \right\} \cap \underline{z}^{(k_i)}$$

$$\to \tilde{z} \quad (i \to \infty).$$

But $\underline{z}^{(k_i+1)} \subseteq \underline{z}^{(k_i)}$ $(\forall i \geq 0)$. So $\underline{z}^{(k)} \to \tilde{z}$ $(k \to \infty)$. This proves (c).

(d) By (c) if $\underline{z}^{(k)} \neq \emptyset$ $(\forall k \geq 0)$ then $\underline{z}^{(k)} \to \tilde{z}$ $(k \to \infty)$ and $f(\tilde{z}) = 0$. Therefore if $\exists z^* \in \hat{\underline{z}}$ such that $f(z^*) = 0$ then it cannot be true that $\underline{z}^{(k)} \neq \emptyset$ $(\forall k \geq 0)$. So for some $k \geq 0$, $\underline{z}^{(k)} = \emptyset$. This proves (d). □

Theorem 5.3.3

Let $f, S, \hat{\underline{z}}$, and $\underline{\hat{D}}$ be as in Theorem 5.3.1. Let $z^* \in \hat{\underline{z}}$ be such that $f(z^*) = 0$. Let $\left(\varsigma^{(k)}\right)$ be a sequence in C which is generated from a point procedure P which is locally convergent to z^* with $O_R(\mathrm{P}, z^*) \geq \nu$. Let the sequences $\left(\underline{z}^{(k)}\right)$ and $\left(z^{(k)}\right)$ be generated from the procedure IP defined by

$$\underline{z}^{(0)} = \hat{\underline{z}}, \qquad 5.3.2a$$

$$z^{(0)} = m_R(\hat{\underline{z}}), \tag{5.3.2b}$$

$$\zeta^{(0)} = z^{(0)}, \tag{5.3.2c}$$

$$\underline{z}^{(k+1)} = \left\{ z^{(k)} - \frac{f(z^{(k)})}{\hat{\underline{D}}} \right\} \cap \underline{z}^{(k)}, \tag{5.3.2d}$$

$$\zeta^{(k+1)} = P(z^{(k)}), \tag{5.3.2e}$$

$$z^{(k+1)} = \kappa(\zeta^{(k+1)}, \underline{z}^{(k+1)}) \quad (k \geq 0), \tag{5.3.2f}$$

in which $\kappa : C \times I_R(C) \rightarrow C$ is defined by

$$\kappa_i(\zeta, \underline{z}) = \begin{cases} z_{iI} & (\zeta_i < z_{iI}) \\ z_{iS} & (z_{iS} < \zeta_i) \quad (i = R, I), \\ \zeta_i & (\text{otherwise}) \end{cases} \tag{5.3.3}$$

where $\underline{z} = \underline{z}_R + i\underline{z}_I$. Then

(a) $z^* \in \underline{z}^{(k+1)} \subseteq \underline{z}^{(k)} \quad (\forall k \geq 0)$;

(b) $\underline{z}^{(k)} \rightarrow z^* \quad (k \rightarrow \infty)$;

(c) $z^{(k)} \rightarrow z^* \quad (k \rightarrow \infty)$;

(d) $O_R(\text{IP}, z^*) \geq \nu$.

Proof

(a) That $z^* \in \underline{z}^{(k+1)} \subseteq \underline{z}^{(k)}$ $(\forall k \geq 0)$ follows from Theorem 5.3.2(a).

(b) That $\underline{z}^{(k)} \to z^*$ $(k \to \infty)$ follows from Theorem 5.3.2(c).

(c) Now $z^{(k)} \in \underline{z}^{(k)}$ $(\forall k \geq 0)$ and by (a), $z^* \in \underline{z}^{(k)}$ $(\forall k \geq 0)$. So by (b), $z^{(k)} \to z^*$ $(k \to \infty)$.

(d) By 5.3.2*d*, Proposition 1.3.2.9(e),(a),(b), and Lemma 5.2.1,

$$w_R(\underline{z}^{(k+1)}) \leq |\underline{\hat{D}}|_R w_R(1/\underline{\hat{D}})|z^{(k)} - z^*|_R$$

$$\leq \alpha|z^{(k)} - z^*|, \qquad 5.3.4$$

where $\alpha = 2|\underline{\hat{D}}|_R w_R(1/\underline{\hat{D}})$. Also by 5.3.3, $|z^{(k)} - z^*| \leq |\zeta^{(k)} - z^*|$, so $R_\nu(z^{(k)}) < 1$, and by 5.3.4

$$w_R(\underline{z}^{(k+1)}) \leq \alpha|\zeta^{(k)} - z^*| \quad (\forall k \geq 0),$$

whence $R_\nu(\underline{z}^{(k)}) < 1$. Therefore $O_R(\mathrm{IP}, z^*) \geq \nu$. □

5.4 Interval Inclusions of Complex Polynomial Zeros

The polynomial $p : C \to C$ defined by 5.1.1 is an entire function and is therefore analytic in any open convex set $S \subseteq C$. If $\hat{\underline{z}} \in I_R(C)$ then it is easy to determine $\underline{\hat{D}} \in I_R(C)$ such that if $p(z) = u(x, y) + iv(x, y)$ then $\partial_1 u(x', y') + i\partial_1 v(x'', y'') \in \underline{\hat{D}}$ $(\forall x', x'' \in \hat{\underline{z}}_R)$ $(\forall y', y'' \in \hat{\underline{z}}_I)$ where $\hat{\underline{z}} = \hat{\underline{z}}_R + i\hat{\underline{z}}_I$, by evaluating $\underline{p}' : I_R(C) \to I_R(C)$ at $\hat{\underline{z}}$, where $\underline{p}'$ is a continuous inclusion monotonic interval extension of the derivative p' of p. For example, if $\hat{\underline{z}} = [-3, -1] + i[1, 3]$ and $p(z) = z^2 + 4z + 5$ then an inclusion monotonic interval extension of p' is given by

$$\underline{p}'(\underline{z}) = 2\underline{z} + 4$$

and

$$\underline{p}'(\hat{\underline{z}}) = 2\hat{\underline{z}} + 4$$

$$= [-2, 2] + i[2, 6].$$

Let $\hat{\underline{z}}_i \in I_R(C)$ be such that $0 \notin \hat{\underline{D}}_i = \underline{p}'(\hat{\underline{z}}_i)$ $(i = 1, \ldots, n)$. Let $\underline{N}_i : C \times I_R(C) \to I_R(C)$ be defined by

$$\underline{N}_i(\zeta, \hat{\underline{z}}_i) = \zeta - \frac{p(\zeta)}{\hat{\underline{D}}_i} \quad (i = 1, \ldots, n). \tag{5.4.1}$$

If $\underline{N}_i(m_R(\hat{\underline{z}}_i), \hat{\underline{z}}_i) \cap \hat{\underline{z}}_i = \emptyset$ for at least one $i \in \{1, \ldots, n\}$ then by Theorem 5.3.1(a) $\nexists z_i^* \in \hat{\underline{z}}_i$ such that $p(z_i^*) = 0$. If $\underline{N}_i(m_R(\hat{\underline{z}}_i), \hat{\underline{z}}_i) \subseteq \hat{\underline{z}}_i$ then by Theorem 5.3.1(b),(c) there exists a unique simple zero of p in $\hat{\underline{z}}_i$.

Suppose that $\hat{\underline{z}}_i \cap \hat{\underline{z}}_j = \emptyset$ $(1 \leq i < j \leq n)$, that $0 \notin \hat{\underline{D}}_i$ $(i = 1, \ldots, n)$, and that $\underline{N}_i(m_R(\hat{\underline{z}}_i), \hat{\underline{z}}_i) \subseteq \hat{\underline{z}}_i$ $(i = 1, \ldots, n)$. Then for $i = 1, \ldots, n$, $\hat{\underline{z}}_i$ contains exactly one simple zero of p.

Let P be a point iterative procedure such as PT1, PS1, PSS1, or PRSS1 (§§ 4.2–4.3) which consists of generating the sequences $\left(\zeta_i^{(k)}\right)$ from

$$\zeta_i^{(k+1)} = P_i(\zeta_1^{(k)}, \ldots, \zeta_n^{(k)}) \quad (i = 1, \ldots, n)\,(k \geq 0).$$

Then the procedure IP for bounding the simple zeros of p consists of generating the sequences

$\left(\underline{z}_i^{(k)}\right)$, $\left(z_i^{(k)}\right)$ $(i = 1, \ldots, n)$ from

$$\underline{z}_i^{(0)} = \hat{\underline{z}}_i, \tag{5.4.2a}$$

$$z_i^{(0)} = m_R(\hat{\underline{z}}_i), \tag{5.4.2b}$$

$$\zeta_i^{(0)} = z_i^{(0)}, \tag{5.4.2c}$$

$$\underline{z}_i^{(k+1)} = \left\{ z_i^{(k)} - \frac{p(z_i^{(k)})}{\hat{\underline{D}}_i} \right\} \cap \underline{z}_i^{(k)}, \tag{5.4.2d}$$

$$\zeta_i^{(k+1)} = P_i(z_1^{(k)}, \ldots, z_n^{(k)}), \tag{5.4.2e}$$

$$z_i^{(k+1)} = \kappa(\zeta_i^{(k+1)}, \underline{z}_i^{(k+1)}) \ (k \geq 0). \tag{5.4.2f}$$

The procedures IPT1, IPS1, IPSS1, IPRSS1 are obtained by using, in IP, the point procedures PT1, PS1, PSS1, and PRSS1 respectively.

The following theorems are immediate consequences of Theorems 4.2.1, 4.2.2, 4.3.1, and 5.3.3.

Theorem 5.4.1

Let $p : C \to C$ be defined by 5.1.1. If (1) p has n simple zeros $z_i^* \in C$ $(i = 1, \ldots, n)$; (2) $\hat{\underline{z}}_i \in I_R(C)$ $(i = 1, \ldots, n)$ are such that $z_i^* \in \hat{\underline{z}}_i$ $(i = 1, \ldots, n)$ and $\hat{\underline{z}}_i \cap \hat{\underline{z}}_j = \emptyset$ $(1 \leq i < j \leq n)$; (3) $\hat{\underline{D}}_i \in I_R(C)$ $(i = 1, \ldots, n)$ are such that $0 \notin \hat{\underline{D}}_i$ $(i = 1, \ldots, n)$ and if

$$p(z) = u(x,y) + iv(x,y) \quad (z = x + iy),$$

then

$$\partial_1 u(x',y') + i\partial_1 v(x'',y'') \in \hat{D}_j \quad (\forall x' + iy', x'' + iy'' \in \hat{z}_j)\,(j = 1,\ldots,n);$$

(4) the sequences $\left(\underline{z}_i^{(k)}\right)$, $\left(\zeta_i^{(k)}\right)$, and $\left(z_i^{(k)}\right)$ $(i = 1,\ldots,n)$ are generated from IPT1, then

(a) $z_i^* \in \underline{z}_i^{(k+1)} \subseteq \underline{z}_i^{(k)} \quad (\forall k \geq 0)\,(i = 1,\ldots,n)$;

(b) $\underline{z}_i^{(k)} \to z_i^* \quad (k \to \infty)\,(i = 1,\ldots,n)$;

(c) $z_i^{(k)} \to z_i^* \quad (k \to \infty)\,(i = 1,\ldots,n)$;

(d) $O_R(\text{IPT1}, z_i^*) \geq 2 \quad (i = 1,\ldots,n)$. □

Theorem 5.4.2

If Hypotheses (1)–(3) of Theorem 5.4.1 are valid; (4) the sequences $\left(\underline{z}_i^{(k)}\right)$, $\left(\zeta_i^{(k)}\right)$, $\left(z_i^{(k)}\right)$ $(i = 1,\ldots,n)$ are generated from IPS1, then conclusions (a)–(c) of Theorem 5.4.1 are valid; (d) $O_R(\text{IPS1}, z_i^*) \geq 1 + \sigma$ where $\sigma \in (1,2)$ is the unique positive zero of $t^n - t - 1$, $(i = 1,\ldots,n)$. □

Theorem 5.4.3

If Hypotheses (1)–(3) of Theorem 5.4.1 are valid; (4) the sequences $\left(\underline{z}_i^{(k)}\right)$, $\left(\zeta_i^{(k)}\right)$, $\left(z_i^{(k)}\right)$ $(i = 1,\ldots,n)$ are generated from IPRSS1 with $m^{(k)} = m$ $(\forall k \geq 0)$, then conclusions (a)–(c) of Theorem 5.4.1 are valid; (d) $O_R(\text{IPRSS1}, z_i^*) \geq 2m + 1$ $(i = 1,\ldots,n)$. □

Recently Petković and Stefanović [PetS--86a] have discussed the interval procedures IT1, IS1, and a procedure, here referred to as IRPS1, which is comparable with a special case of the procedure IRSS1 for simultaneously bounding simple polynomial zeros using rectangular complex interval arithmetic. The procedures IT1, IS1, IRPS1, and IRSS1 consist

of generating the sequences $\left(\underline{z}_i^{(k)}\right)$ $(i = 1, \ldots, n)$ as follows.

IT1 :

$$\underline{z}_i^{(k+1)} = \left\{ z_i^{(k)} - \frac{p(z_i^{(k)})}{\prod_{\substack{j=1 \\ j \neq i}}^{n} (z_i^{(k)} - \underline{z}_j^{(k)})} \right\} \cap \underline{z}_i^{(k)} \quad (i = 1, \ldots, n)(k \geq 0). \tag{5.4.3}$$

IS1 :

$$\underline{z}_i^{(k+1)} = \left\{ z_i^{(k)} - \frac{p(z_i^{(k)})}{\prod_{j=1}^{i-1} (z_i^{(k)} - \underline{z}_j^{(k+1)}) \prod_{j=i+1}^{n} (z_i^{(k)} - \underline{z}_j^{(k)})} \right\} \cap \underline{z}_i^{(k)} \; (i = 1, \ldots, n)(k \geq 0). \tag{5.4.4}$$

IRPS1 :

$$\underline{z}_i^{(k,1)} = \left\{ z_i^{(k)} - \frac{p(z_i^{(k)})}{\prod_{j=1}^{i-1} (z_i^{(k)} - \underline{z}_j^{(k,1)}) \prod_{j=i+1}^{n} (z_i^{(k)} - \underline{z}_j^{(k)})} \right\} \cap \underline{z}_i^{(k)} \quad (i = 1, \ldots, n), \tag{5.4.5a}$$

$$\underline{z}_i^{(k,2)} = \left\{ z_i^{(k)} - \frac{p(z_i^{(k)})}{\prod_{j=1}^{i-1} (z_i^{(k)} - \underline{z}_j^{(k,2)}) \prod_{j=i+1}^{n} (z_i^{(k)} - \underline{z}_j^{(k,1)})} \right\} \cap \underline{z}_i^{(k,1)} \quad (i = 1, \ldots, n), \tag{5.4.5b}$$

$$\underline{z}_i^{(k+1)} = \underline{z}_i^{(k,2)} \quad (i = 1, \ldots, n)\,(k \geq 0). \tag{5.4.5c}$$

IRSS1 :

$$\underline{z}_i^{(k,0)} = \underline{z}_i^{(k)} \quad (i = 1, \ldots, n), \tag{5.4.6a}$$

$$\underline{z}_i^{(k,2l-1)} = \left\{ z_i^{(k)} - \frac{p(z_i^{(k)})}{\prod_{j=1}^{i-1}(z_i^{(k)} - \underline{z}_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(z_i^{(k)} - \underline{z}_j^{(k,2l-2)})} \right\} \cap \underline{z}_i^{(k,2l-2)} \qquad 5.4.6b$$

$$(i = 1, \ldots, n),$$

$$\underline{z}_i^{(k,2l)} = \left\{ z_i^{(k)} - \frac{p(z_i^{(k)})}{\prod_{j=1}^{i-1}(z_i^{(k)} - \underline{z}_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(z_i^{(k)} - \underline{z}_j^{(k,2l)})} \right\} \cap \underline{z}_i^{(k,2l-1)} \qquad 5.4.6c$$

$$(i = n, \ldots, 1),$$

$$\left(l = 1, \ldots, m^{(k)}\right),$$

$$\underline{z}_i^{(k+1)} = \underline{z}_i^{(k,2m^{(k)})} \quad (i = 1, \ldots, n)(k \geq 0), \qquad 5.4.6d$$

where in all procedures, $z_i^{(k)} = m_R(\underline{z}_i^{(k)})$ $(i = 1, \ldots, n)$.

The procedure IRPS1 has R-order at least 3 [PetS--86a]. The procedure IRSS1 with $m^{(k)} = m$ $(\forall k \geq 0)$ has R-order at least $2m + 1$ (§4.5). So if $m^{(k)} = 1$ $(\forall k \geq 0)$ then the R-order of IRSS1 is at least 3, and IRSS1 has the advantages over IRPS1 that the first sum in the denominator of 5.4.6*b* can be re-used in 5.4.6*c*, that $\underline{z}_n^{(k,2l)} = \underline{z}_n^{(k,2l-1)}$ $(l \geq 1)$, and that $\underline{z}_1^{(k,2l+1)} = \underline{z}_1^{(k,2l)}$ $(l \geq 1)$.

5.5 Numerical Results

The procedures IT1, IS1, ISS1, IRPS1, IRSS1, IPT1, IPS1, IPSS1, IPRPS1 and IPRSS1 have been implemented in Triplex S-algol [ColM--82] [McbWs-83] using software for rectangular complex interval arithmetic as defined in [RokL--71], [RokL--75], and [AleH--83]

on a VAX 11–785 computer. Numerical results are presented for 6 examples (Appendix A), in which $p : C \to C$ is defined by 5.1.1 and has zeros z_i^* $(i = 1, \ldots, n)$. The complex number $x + iy$ is written (x, y) and ke^n denotes $k \times 10^n$.

The initial intervals for each example satisfy the hypotheses of Theorem 5.3.3 and the stopping criterion is

$$\max_{1 \le i \le n} \{w_R(z_i^{(k)})\} \le 10^{-10}.$$

Tables 5.5.1a–5.5.1c contain the cpu times in seconds required by IT1 and IPT1, IS1 and IPS1, ISS1 and IPSS1 respectively. The integers in parentheses are the numbers of iterations required to satisfy the stopping criterion. Table 5.5.1d contains the cpu times for the procedure IRSS1 with automatic determination of $m^{(k)}$ as described in §4.6, and the cpu times for the procedure IPRSS1 with $m^{(k)} = 3$ $(\forall k \ge 0)$. Even better results are obtained for Examples 5.1, 5.5, and 5.6 (Appendix A) using IPRSS1 with $m^{(k)} = 6$ $(\forall k \ge 0)$, and with $m^{(k)} = 6$ $(\forall k \ge 0)$, the cpu time for Example 5.2 using IPRSS1 is 3.9 seconds.

Table 5.5.2 contains the cpu time in seconds required by IRPS1, ISS1, IPRPS1 and IPSS1, where IPRPS1 is the interval version of the point procedure PRPS1. The point procedure PRPS1 corresponding to IRPS1 is given by

$$x_i^{(k,1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k)})}$$

$$x_i^{(k,2)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,2)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k,1)})} \qquad (i = 1, \ldots, n),$$

$$(i = 1, \ldots, n),$$

$$x_i^{(k+1)} = x_i^{(k,2)} \quad (i = 1, \ldots, n)(k \geq 0).$$

Table 5.5.2 suggests that the procedure ISS1 requires less cpu times then does IRPS1 when both methods converge with the same number of iterations. This is also true for the procedures IPSS1 and IPRPS1. Clearly, four examples out of six are in favour of ISS1 and IPSS1. Therefore the procedure SS1 usually is more efficient then RPS1.

Suppose that procedures are ranked in efficiency by using the relation $\leq$ defined by $(P_1 \leq P_2) \Leftrightarrow (T_1 \geq T_2)$, where T_1 and T_2 are the cpu times required by procedures P_1 and P_2 respectively to satisfy a given stopping criterion with given initial data. Then the results in Tables 5.5.1a–5.5.1d and other results which are not reported in this thesis suggest that IT1<IPT1, IS1<IPS1, ISS1<IPSS1,IRSS1<IPRSS1, IT1<IS1$\leq$IRSS1, IPT1<IPSS1<IPRSS1, and IPS1<IPSS1. Also Table 5.5.2 suggests that IRPS1<IPRPS1, IRPS1<ISS1 and IPRPS1<IPSS1.

These results suggest that of the procedures IT1, IS1, ISS1, IRSS1, IPT1, IPS1, IPSS1, IPRSS1, the procedure IPRSS1 with $m^{(k)} = 3$ $(\forall k \geq 0)$ is likely to be the most efficient for bounding the simple zeros of a complex polynomial.

Example	IT1	IPT1
5.1	17.40(6)	8.64(4)
5.2	14.43(5)	9.02(4)
5.3	21.37(4)	12.25(3)
5.4	21.46(4)	11.99(3)
5.5	9.70(6)	5.21(4)
5.6	8.73(5)	5.72(4)

Table 5.5.1a

Example	IS1	IPS1
5.1	14.72(5)	8.69(4)
5.2	11.47(4)	8.90(4)
5.3	21.35(4)	12.21(3)
5.4	21.62(4)	12.05(3)
5.5	8.04(5)	5.25(4)
5.6	6.91(4)	5.44(4)

Table 5.5.1b

Example	ISS1	IPSS1
5.1	16.72(4)	6.94(3)
5.2	12.20(3)	5.32(2)
5.3	22.17(3)	9.06(2)
5.4	22.06(3)	9.05(2)
5.5	6.92(3)	4.07(3)
5.6	7.50(3)	4.34(3)

Table 5.5.1c

Example	IRSS1	IPRSS1
5.1	15.37(3)	5.42(2)
5.2	11.01(2)	5.83(2)
5.3	19.71(2)	5.96(1)
5.4	19.70(2)	6.01(1)
5.5	7.29(3)	3.15(2)
5.6	7.75(3)	3.27(2)

Table 5.5.1d

Example	IRPS1	ISS1	IPRPS1	IPSS1
5.1	12.22(3)	14.30(4)	4.42(2)	6.07(3)
5.2	12.10(3)	11.04(3)	4.64(2)	4.62(2)
5.3	22.12(3)	19.12(3)	8.08(2)	7.88(2)
5.4	14.73(2)	19.37(3)	8.01(2)	7.77(2)
5.5	6.86(3)	6.05(3)	2.57(2)	3.64(3)
5.6	7.09(3)	6.22(3)	3.60(3)	3.57(3)

Table 5.5.2

5.6 Future Work

Computational experience indicates that the repeated symmetric single-step idea for estimating and for bounding the zeros of polynomials simultaneously can be applied to other procedures which have been discussed in §4.2. This will be the subject of future research. The following iterative procedures are of interest in this respect.

PRSS2 :

$$p_i^{(k)} = p(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$\delta_i^{(k)} = p(x_i^{(k)})/q'(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$x_i^{(k,1)} = x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k)} + \delta_j^{(k)})} \quad (i = 1, \ldots, n),$$

$$x_i^{(k,2)} = x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k,2)})} \quad (i = n, \ldots, 1),$$

$$x_i^{(k,2l-1)} = x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k,2l-2)})} \quad (i = 1, \ldots, n),$$

$$x_i^{(k,2l)} = x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - x_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - x_j^{(k,2l)})} \quad (i = n, \ldots, 1),$$

$$\left(l = 2, \ldots, r^{(k)}\right),$$

$$x_i^{(k+1)} = x_i^{(k,2r^{(k)})} \quad (i = 1, \ldots, n)(k \geq 0),$$

where $q'(x)$ is given by 4.2.6.

PRSS3 :

$$x_i^{(k,0)} = x_i^{(k)} \quad (i = 1, \ldots, n),$$

$$g_i^{(k)} = p(x_i^{(k)})/p'(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$x_i^{(k,2l-1)} = x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)}\left\{\sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - x_j^{(k,2l-1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - x_j^{(k,2l-2)})}\right\}\right]}$$

$$(i = 1, \ldots, n),$$

$$x_i^{(k,2l)} = x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)}\left\{\sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - x_j^{(k,2l-1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - x_j^{(k,2l)})}\right\}\right]}$$

$$(i = n, \ldots, 1),$$

$$\left(l = 1, \ldots, r^{(k)}\right),$$

$$x_i^{(k+1)} = x_i^{(k,2r^{(k)})} \quad (i = 1, \ldots, n)(k \geq 0).$$

PRSS4 :

$$g_i^{(k)} = p(x_i^{(k)})/p'(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$x_i^{(k,1)} = x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)}\left\{\sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - x_j^{(k,1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - x_j^{(k)} + g_j^{(k)})}\right\}\right]}$$

$$(i = 1, \ldots, n),$$

$$x_i^{(k,2)} = x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)}\left\{\sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - x_j^{(k,1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - x_j^{(k,2)})}\right\}\right]}$$

$$(i = n, \ldots, 1),$$

$$x_i^{(k,2l-1)} = x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)}\left\{\sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - x_j^{(k,2l-1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - x_j^{(k,2l-2)})}\right\}\right]}$$

$$(i = 1, \ldots, n),$$

$$x_i^{(k,2l)} = x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)}\left\{\sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - x_j^{(k,2l-1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - x_j^{(k,2l)})}\right\}\right]}$$

$$(i = n, \ldots, 1),$$

$$\left(l = 2, \ldots, r^{(k)}\right),$$

$$x_i^{(k+1)} = x_i^{(k,2r^{(k)})} \quad (i = 1, \ldots, n)(k \geq 0).$$

The corresponding interval procedures are as follows.

IRSS2 :

$$\underline{d}_i = \underline{p}'(\underline{x}_i^{(0)}) \quad (i = 1, \ldots, n),$$

$$x_i^{(k)} = m(\underline{x}_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$p_i^{(k)} = p(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$\underline{z}_i^{(k)} = \left\{x_i^{(k)} - \frac{p_i^{(k)}}{\underline{d}_i}\right\} \cap \underline{x}_i^{(k)} \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,1)} = \left\{x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{z}_j^{(k)})}\right\} \cap \underline{z}_i^{(k)} \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2)})} \right\} \cap \underline{x}_i^{(k,1)} \quad (i = n, \ldots, 1),$$

$$\underline{x}_i^{(k,2l-1)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l-2)})} \right\} \cap \underline{x}_i^{(k,2l-2)} \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l)})} \right\} \cap \underline{x}_i^{(k,2l-1)} \quad (i = n, \ldots, 1),$$

$$\left(l = 2, \ldots, m^{(k)}\right),$$

$$\underline{x}_i^{(k+1)} = \underline{x}_i^{(k,2m^{(k)})} \quad (i = 1, \ldots, n)(k \geq 0).$$

IRSS3 :

$$\underline{x}_i^{(k,0)} = \underline{x}_i^{(k)} \quad (i = 1, \ldots, n),$$

$$x_i^{(k)} = m(\underline{x}_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$g_i^{(k)} = p(x_i^{(k)})/p'(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l-1)} = \left\{ x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)}\left\{\sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(k,2l-1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(k,2l-2)})}\right\}\right]} \right\} \cap \underline{x}_i^{(k,2l-2)}$$

$$(i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l)} = \left\{ x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)} \left\{ \sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(k,2l-1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(k,2l)})} \right\}\right]} \right\} \cap \underline{x}_i^{(k,2l-1)}$$

$$(i = n, \ldots, 1),$$

$$\left(l = 1, \ldots, m^{(k)}\right),$$

$$\underline{x}_i^{(k+1)} = \underline{x}_i^{(k,2m^{(k)})} \quad (i = 1, \ldots, n)(k \geq 0).$$

IRSS4 :

$$\underline{d}_i = \underline{p}'(\underline{x}_i^{(0)}) \quad (i = 1, \ldots, n),$$

$$x_i^{(k)} = m(\underline{x}_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$g_i^{(k)} = p(x_i^{(k)}) / p'(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$\underline{z}_i^{(k)} = \left\{ x_i^{(k)} - \frac{p(x_i^{(k)})}{\underline{d}_i} \right\} \cap \underline{x}_i^{(k)} \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,1)} = \left\{ x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)} \left\{ \sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(k,1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - \underline{z}_j^{(k)})} \right\}\right]} \right\} \cap \underline{z}_i^{(k)}$$

$$(i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2)} = \left\{ x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)} \left\{ \sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(k,1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(k,2)})} \right\}\right]} \right\} \cap \underline{x}_i^{(k,1)}$$

$$(i = n, \ldots, 1),$$

$$\underline{x}_i^{(k,2l-1)} = \left\{ x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)} \left\{ \sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(2l-1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(2l-2)})} \right\} \right]} \right\} \cap \underline{x}_i^{(2l-2)}$$

$$(i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l)} = \left\{ x_i^{(k)} - \frac{g_i^{(k)}}{\left[1 - g_i^{(k)} \left\{ \sum_{j=1}^{i-1} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(2l-1)})} + \sum_{j=i+1}^{n} \frac{1}{(x_i^{(k)} - \underline{x}_j^{(2l)})} \right\} \right]} \right\} \cap \underline{x}_i^{(k,2l-1)}$$

$$(i = n, \ldots, 1),$$

$$\left(l = 2, \ldots, m^{(k)}\right),$$

$$\underline{x}_i^{(k+1)} = \underline{x}_i^{(k,2m^{(k)})} \quad (i = 1, \ldots, n)(k \geq 0).$$

The second generalization of the repeated symmetric single-step procedures would be, for example, for IRSS1

$$\underline{x}_i^{(k,0)} = \underline{x}_i^{(k)} \quad (i = 1, \ldots, n),$$

$$x_i^{(k)} = m(\underline{x}_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$p_i^{(k)} = p(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l-1)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1} (x_i^{(k)} - \underline{x}_j^{(k,2l-1)}) \prod_{j=i+1}^{n} (x_i^{(k)} - \underline{x}_j^{(k,2l-2)})} \right\} \cap \underline{x}_i^{(k,2l-2)} \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l)})} \right\} \cap \underline{x}_i^{(k,2l-1)} \quad (i = n, \ldots, 1),$$

$$\underline{x}_i^{(k,2l+1)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l+1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l)})} \right\} \cap \underline{x}_i^{(k,2l)} \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l+2)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l+1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l+2)})} \right\} \cap \underline{x}_i^{(k,2l+1)} \quad (i = n, \ldots, 1),$$

$$\left(l = 1, \ldots, m^{(k)}\right),$$

$$\underline{x}_i^{(k+1)} = \underline{x}_i^{(k,2m^{(k)}+2)} \quad (i = 1, \ldots, n)(k \geq 0).$$

Finally, the procedure IRPS1 [PetS--86a] can also be generalized as follows.

$$\underline{x}_i^{(k,0)} = \underline{x}_i^{(k)} \quad (i = 1, \ldots, n),$$

$$x_i^{(k)} = m(\underline{x}_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$p_i^{(k)} = p(x_i^{(k)}) \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l-1)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l-1)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l-2)})} \right\} \cap \underline{x}_i^{(k,2l-2)} \quad (i = 1, \ldots, n),$$

$$\underline{x}_i^{(k,2l)} = \left\{ x_i^{(k)} - \frac{p_i^{(k)}}{\prod_{j=1}^{i-1}(x_i^{(k)} - \underline{x}_j^{(k,2l)}) \prod_{j=i+1}^{n}(x_i^{(k)} - \underline{x}_j^{(k,2l-1)})} \right\} \cap \underline{x}_i^{(k,2l-1)} \quad (i = 1, \ldots, n),$$

$\left(l = 1, \ldots, m^{(k)}\right)$,

$\underline{x}_i^{(k+1)} = \underline{x}_i^{(k,2m^{(k)})} \quad (i = 1, \ldots, n)(k \geq 0)$.

APPENDIX A

Example Problems

This appendix contains several examples which are used to compare the algorithms presented in this thesis.

Example 3.1 [Sis---82a]:

Oren's power function;

$$f(x_1, \ldots, x_n) = \left(\sum_{i=1}^{n} i x_i^2\right)^2.$$

Starting point: $(1, \ldots, 1)$

Example 3.2 [Sis---82a]:

Extended Rosenbrock function;

$$f(x_1, \ldots, x_n) = \sum_{i=2}^{n} \left(100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2\right)$$

Starting point: $(-1.2, 1, -1.2, 1, \ldots, -1.2, 1)$

Example 3.3 [Sis---82a]:

Powell's quartic function;

$$f(x_1, \ldots, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

Starting point: $(3, -1, 0, 1)$

Example 3.4 [Sis---82a]:

Rosenbrock's function;

$$f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$$

Starting point: $(-1.2, 1)$

Example 3.5 [Sis---82a]:

Rosenbrock's cliff function;

$$f(x_1, x_2) = (0.01(x_1 - 3))^2 - (x_1 - x_2) + exp(20(x_1 - x_2))$$

Starting point: $(0, -1)$

Example 3.6 [Sis---82a]:

Hyperbola-circle function;

$$f(x_1, x_2) = (x_1 x_2 - 1)^2 + (x_1^2 + x_2^2 - 4)^2$$

Starting point: $(0, 1)$

Example 3.6 [Sis---82a]:

Sisser's function:

$$f(x_1, x_2) = 3x_1^4 - 2x_1^2 x_2^2 + 3x_2^4$$

Starting point: $(1, 0.1)$

Example 4.1 [AleH--83]:

The characteristic polynomial

$$p(\lambda) = det(\lambda I - A), \qquad A.1(a)$$

where

$$A = \begin{pmatrix} a_1 & b_1 & & & \bigcirc \\ b_1 & a_2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b_{n-1} \\ \bigcirc & & & b_{n-1} & a_n \end{pmatrix} \qquad A.1(b)$$

and

$$\left.\begin{aligned} f^{(0)}(\lambda) &= 1, \\ f^{(1)}(\lambda) &= (\lambda - a_k), \\ f^{(k)}(\lambda) &= (\lambda - a_k) f^{(k-1)}(\lambda) - (b_{k-1})^2 f^{(k-2)}(\lambda) \quad 2 \le k \le n, \\ p(\lambda) &= f^{(n)}(\lambda). \end{aligned}\right\} \qquad A.1(c)$$

For this example [AleH--83]:

$$n = 9,$$

$$b_i = 1 \quad (i = 1, \ldots, n),$$

$$a_1 = 15; a_2 = 10; a_3 = 7; a_4 = 4,$$

$$a_5 = 0; a_6 = -4; a_7 = -7; a_8 = -10; a_9 = -15$$

Initial intervals:

$$\underline{x}_1^{(0)} = [14, 16], \quad \underline{x}_2^{(0)} = [8, 12], \quad \underline{x}_3^{(0)} = [5, 9],$$
$$\underline{x}_4^{(0)} = [2, 6], \quad \underline{x}_5^{(0)} = [-2, 2], \quad \underline{x}_6^{(0)} = [-6, -2],$$
$$\underline{x}_7^{(0)} = [-9, -5], \quad \underline{x}_8^{(0)} = [-12, -8], \quad \underline{x}_9^{(0)} = [-17, -12].$$

Example 4.2 [AleH--83]:

The polynomial is given by $A.1$ with

$$n = 5,$$

$$a_1 = 12, a_2 = 9, a_3 = 6, a_4 = 3, a_5 = 0,$$

$$b_i = 1 \quad (i = 1, \ldots, 4),$$

Initial intervals:

$$\underline{x}_1^{(0)} = [11, 13], \qquad \underline{x}_2^{(0)} = [7, 11], \qquad \underline{x}_3^{(0)} = [4, 8],$$

$$\underline{x}_4^{(0)} = [1, 5], \qquad \underline{x}_5^{(0)} = [-1, 1].$$

Example 4.3 [Wol---86b]:

The polynomial is given by $A.1$ with

$n = 9$,

$a_i = 10$ $(i = 1, \ldots, 9)$,

$b_i = 20$ $(i = 1, \ldots, 8)$,

The zeros: $x_i^* = 10 + 40cos(\frac{i\pi}{n+1})$ $(i = 1, \ldots, n)$.

Initial intervals: $\underline{x}_i^{(0)} = [x_i^* - 2.8, x_i^* + 5.6]$ $(i = 1, \ldots, n)$.

Example 4.4 [Wol---86b]:

The polynomial is as in Example 4.3 save that in this example, $a_{ii} = -10$ $(i = 1, \ldots, n)$.

Example 4.5:

The polynomial is

$$p(x) = \prod_{i=1}^{n}(x - x_i^*).$$

The zeros:

$$x_i^* = \begin{cases} -2(\frac{n}{2} - i + 1) & (i = 1, \ldots, \frac{n}{2}), \\ \\ -x_{n-i+1}^* & (i = \frac{n}{2} + 1, \ldots, n). \end{cases}$$

Initial intervals: $\underline{x}_i^{(0)} = [x_i^* - 0.5, x_i^* + 1.0]$ $(i = 1, \ldots, n)$.

Example 5.1 [Pet---82]:

The polynomial is

$$p(z) = z^7 + z^5 - 10z^4 - z^3 - z + 10,$$

The zeros:

$$z_1^* = (2,0), \qquad z_2^* = (1,0), \qquad z_3^* = (-1,0),$$
$$z_4^* = (0,1), \qquad z_5^* = (0,-1), \qquad z_6^* = (-1,2),$$
$$z_7^* = (-1,-2).$$

Initial rectangles:

$$\underline{z}_j^{(0)} = [z_{jR}^* - 8.7e^{-2}, z_{jR}^* + 0.17] + i[z_{jI}^* - 8.7e^{-2}, z_{jI}^* + 0.17] \quad (j = 1, \ldots, 7).$$

Example 5.2 [Rok---73]:

The polynomial is

$$p(z) = z^7 + (29.5, 11)z^6 + (194, 204.5)z^5 + (465.5, 74)z^4 + (3655, 3000.5)z^3 +$$
$$(4330, -7245)z^2 + (-3150, 2850)z$$

The zeros:

$$z_1^* = (-20,-1), \qquad z_2^* = (-10,-10), \qquad z_3^* = (0,-5),$$
$$z_4^* = (0,3), \qquad z_5^* = (0,2), \qquad z_6^* = (0.5,0),$$
$$z_7^* = (0,0).$$

Initial rectangles:

$$\underline{z}_j^{(0)} = [z_{jR}^* - 7.8e^{-2}, z_{jR}^* + 0.16] + i[z_{jI}^* - 7.8e^{-2}, z_{jI}^* + 0.16] \quad (j = 1, \ldots, 7).$$

Example 5.3 [Pet---81]:

The polynomial is

$$p(z) = z^{10} + (0,-5)z^9 + (-6,0)z^8 + (-1,0)z^2 + (0,5)z + (6,0)$$

The zeros:

$$z_1^* = (1,0), \qquad z_2^* = (-1,0), \qquad z_3^* = (0,1),$$
$$z_4^* = (0,-1), \qquad z_5^* = (\tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}), \qquad z_6^* = (\tfrac{1}{\sqrt{2}}, -\tfrac{1}{\sqrt{2}}),$$
$$z_7^* = (-\tfrac{1}{\sqrt{2}}, \tfrac{1}{\sqrt{2}}), \qquad z_8^* = (-\tfrac{1}{\sqrt{2}}, -\tfrac{1}{\sqrt{2}}), \qquad z_9^* = (0,2),$$
$$z_{10}^* = (0,3).$$

Initial rectangles:

$$\underline{z}_j^{(0)} = [z_{jR}^* - 1.25e^{-2}, z_{jR}^* + 2.5e^{-2}] + i[z_{jI}^* - 1.25e^{-2}, z_{jI}^* + 2.5e^{-2}] \quad (j = 1, \ldots, 10).$$

Example 5.4 [WanW--87]:

The polynomial is

$$p(z) = z^{10} + (-20,-20)z^9 + (0,4e^2)z^8 + (3e^4,0)z^6 + (-6e^5,-6e^5)z^5 + (0,12e^6)z^4 + (-4e^8,0)z^2 + (8e^9,8^9)z + (0,-16e^{10})$$

The zeros:

$$z_1^* = (10,0), \qquad z_2^* = (-10,0), \qquad z_3^* = (0,10),$$
$$z_4^* = (0,-10), \qquad z_5^* = (10,10), \qquad z_6^* = (10,-10),$$
$$z_7^* = (-10,10), \qquad z_8^* = (-10,-10), \qquad z_9^* = (20,0),$$
$$z_{10}^* = (0,20).$$

Initial rectangles:

$$\underline{z}_j^{(0)} = [z_{jR}^* - 0.125, z_{jR}^* + 0.25] + i[z_{jI}^* - 0.125, z_{jI}^* + 0.25] \quad (j = 1, \ldots, 10).$$

Example 5.5 [PetS--86]:

The polynomial is

$$p(z) = z^5 + (-4,-5)z^4 + (6,20)z^3 + (-4,-30)z^2 + (-15,20)z + (0,75)$$

The zeros:

$$z_1^* = (1,2), \qquad z_2^* = (1,-2), \qquad z_3^* = (-1,0),$$
$$z_4^* = (3,0), \qquad z_5^* = (0,5).$$

Initial rectangles:

$$\underline{z}_j^{(0)} = [z_{jR}^* - 0.25, z_{jR}^* + 0.5] + i[z_{jI}^* - 0.25, z_{jI}^* + 0.5] \quad (j = 1,\ldots,5).$$

Example 5.6 [Rok---73]:

The polynomial is

$$p(z) = z^5 + (-15,-15)z^4 + (0,170)z^3 + (450,-450)z^2 + (-1096,0)z + (480,480)$$

The zeros:

$$z_1^* = (1,1), \qquad z_2^* = (2,2), \qquad z_3^* = (3,3),$$
$$z_4^* = (4,4), \qquad z_5^* = (5,5).$$

Initial rectangles:

$$\underline{z}_j^{(0)} = [z_{jR}^* - 0.125, z_{jR}^* + 0.25] + i[z_{jI}^* - 0.125, z_{jI}^* + 0.25] \quad (j = 1,\ldots,5).$$

REFERENCES

[Abe---73] **Aberth,O.,** *Iteration methods for finding all zeros of a polynomial simultaneously,* Maths. of Comput. **27**(1973)339–344.

[Ait---50] **Aitken,A.C.,** *Studies in practical mathematics V. On the iterative solution of linear equation,* Proc. Roy. Soc. Edinburgh Sec. A **63**(1950)52–60.

[Ale---77] **Alefeld,G.,** *The symmetric single-step method for systems of simultaneous linear equations with intervals as coefficients,* Computing **18**(1977)329–340, (in German).

[AleH--74] **Alefeld,G., Herzberger,J.,** *On the convergence speed of some algorithms for the simultaneous approximation of polynomial roots,* SIAM J. Numer. Anal. **11**(1974)237–243.

[AleH--83] **Alefeld,G., Herzberger,J.,** *Introduction to Interval Computations,* Academic Press, New York, 1983.

[AleP--83] **Alefeld,G., Platzöder,L.,** *A quadratically convergent Krawczyk-like algorithm,* SIAM J. Numer. Anal. **20**(1983)210–219.

[BaiCM-82] **Bailey,P.J., Cole,A.J., Morrison,R.,** *Triplex User Manual CS/82/5,* Department of Computational Science, University of St. Andrews, Fife, KY16 9SS Scotland, 1982.

[Bog---77] **Bogen,R.A., *et al.*,** *MACSYMA Reference Manual,* MIT Laboratory for Computer Science, Cambridge, Mass., 1977.

[Bor---63] **Borsch-Supan,W.,** *A posteriori error bounds for the zeros of polynomials,* Numer. Math. **5**(1963)380–398.

[Bor---70] **Borsch-Supan,W.,** *Residue estimation for polynomial zeros by means of lagrange interpolation,* Numer. Math. **14**(1970)287–296, (in German).

[BraH--73] **Braess,D., Hadeler,K.P.,** *Simultaneous inclusion of the zeros of a polynomial,* Numer. Math. **21**(1973)161–165.

[Cap---78] **Caprani,O., Madsen,K.,** *Iterative methods for interval inclusion of fixed points*. BIT **18**(1978)42–51.

[Chu---60] **Churchill,R.V.,** *Complex Variables and Applications,* Mc Graw-Hill Kogakusha Ltd., Tokyo, 1960.

[ColM--82a] **Cole,A.J., Morrison,R.,** *An Introduction to Programming with S-algol,* Cambrage University Press, Cambrage, London, New York, New Rochelle, Melbourne, Sydney, 1982.

[ColM--82b] **Cole,A.J., Morrison,R.,** *Triplex: A system for interval arithmetic,* Software—Practice and Experience **12**(1982)341–350.

[DenS--83] **Dennis,J.E., Schnabel,R.E.,** *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Prentice-Hall Series in Computational Mathematics, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1983.

[Dur---60] **Durand,E.,** *Solution numérique des eéquations algébraique* (tome 1), Masson, Paris, 1960.

[Ehr---67] **Ehrlich,L.W.,** *A modified Newton method for polynomials,* Comm. ACM **10**(1967)107–108.

[FarL--75] **Farmer,M.R., Loizou,G.,** *A class of iteration functions for improving, simultaneously, approximations to the zeros of a polynomial,* BIT **15**(1975) 250–258.

[FarL--77] **Loizou,G.,** *An algorithm for the total, or partial, factorization of a polynomial,* Math. Proc. Cambridge Phil. Soc. **82**(1977)427–437.

[Gar---75] **Gargantini,I.,** *Parallel square root iterations,* Interval Mathematics (K. Nickel Ed.) Lecture Notes in Computer Science **29** Springer Verlag, Heidelberg, 1975.

[Gar---76] **Gargantini,I.,** *Parallel Laguerre iterations: The complex case,* Numer. Math. **26**(1976)317–323.

[Gar---78] **Gargantini,I.,** *Further applications of circular arithmetic : Schroeder-like algorithms with error bounds for finding zeros of polynomials,* SIAM J. Numer. Anal. **15**(1978)497–510.

[Gar---81] **Gargantini,I.,** *An application of interval mathematics: A polynomial solver with degree four convergence,* Freiburger Intervallbericht **81/7**, 1981.

[GarH--72] **Gargantini,I., Henrici,P.,** *Circular arithmetic and the determination of polynomial zeros,* Numer. Math. **18**(1972)305–320.

[Gla---75] **Glatz,G.,** *Newton algorithms for the determination of polynomial roots using complex circular arithmetic,* Interval Mathematics (K. Nickel Ed.) Lecture Notes in Computer Science **29**, Springer Verlag, Heidelberg, 1975, (in German).

[Han---78a] **Hansen,E.,** *Interval forms of Newton's method.* Computing **20**(1978)153–163.

[Han---78b] **Hansen,E.** *A globally convergent interval method for computing and bounding real roots.* BIT **18**(1978)415–424.

[HaPR--77] **Hansen,E., Patrick,M., Rusnak,J.,** *Some modifications of Laguerre's method,* BIT **17** (1977)409–417.

[Hen---70] **Henrici,P.,** *Methods of search for solving polynomial polynomial equations,* J. ACM **17**(1970)273–283.

[Hen---74] **Henrici,P.,** *Applied and Computational Complex Analysis,* Vol. **1**, John Wiley and Sons, New York, 1974.

[Ker---66] **Kerner,O.,** *Total step procedure for the calculation of the zeros of polynomials,* Numer. Math. **8**(1966)290–294, (in German).

[Kje---84] **Kjellberg,G.,** *Two observations on Durand-Kerner's root-finding method,* BIT **24**(1984)556–559.

[KriS--75] **Krier,N., Spellucci,P.,** *Inclusion sets of polynomial zeros,* Interval Mathematics (K. Nickel Ed.) Lecture Notes in Computer Science **29**, Springer Verlag, Heidelberg, 1975, (in German).

[Loi---83] **Loizou,G.,** *Higher-order iteration functions for simultaneously approximating polynomial zeros,* Intern. J. Computer Math. **14**(1983)45–58.

[Mae---54] **Maehly,J.,** *On the iterative solution of algebraic equations*, ZAMP **5**(1954) 260–263, (in German).

[McbWs-83] **Morrison,R., Cole,A.J., Bailey,P., Wolfe,M.A., Shearer,J.M.,** *Experience in using a high level language which supports interval arithmetic,* in : T.R.N. Rao, P. Kornerup Eds., ARITH6, 6th Symposium on Computer Arithmetic, Aarhus, Denmark, (1983)1–16.

[McC---83] **McCormick,G.P.,** *Nonlinear Programming—Theory, Algorithms and Applications,* John Wiley and Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1983.

[MilP--83] **Milovanovic,G.V., Petkovic,M.S.,** *On the convergence of a modified method for simultaneous finding of polynomial zeros,* Computing **30** (1973)171–178.

[Mir---79] **Mirnia-Harikandi,K.,** *Modifications of some Algorithms for Unconstrained Optimization,* Ph.D. Thesis, University of St. Andrews, 1979.

[Moo---62] **Moore,R.E.,** *Interval Arithmetic and Automatic Error Analysis in Digital Computing,* Ph.D Thesis, Stanford University, 1962.

[Moo---79] **Moore,R.E.,** *Methods and Applications of Interval Analysis,* SIAM Publications, Philadelphia, 1979.

[MooQ--82] **Moore,R.E, Qi,L.,** *A successive test for nonlinear systems,* SIAM, J. Numer. Anal., **19**(1982)845–850.

[Neu---84] **Neumaier,A.,** *An interval version of the secant method,* BIT **24**(1984)366–372.

[Neu---85] **Neumaier,A.,** *Interval iteration for zeros of systems of equations,* BIT **25**(1985)256–273.

[Nou---75] **Nourein,A.W.,** *An iteration formula for the simultaneous determination of the zeros of a polynomial,* J. Comput. and Appl. Math. **1**(1975)251–254.

[Nou---77a] **Nourein,A.W.,** *An improvement on Nourein's method for the simultaneous determination of zeros of a polynomial (an algorithm),* J. Comput. and Appl. Math. **3**(1977)109–112.

[Nou---77b] **Nourein,A.W.,** *An improvement on two iteration methods for simultaneous determination of the zeros of a polynomial,* Intern. J. Computer Math. Sect. B **6**(1977)241–252.

[OrtR--70] **Ortega,J.M., Rheinboldt,W.C.,** *Iterative Solution of Nonlinear Equations in Several Variables,* Academic Press, New York, 1970.

[Pet---80] **Petkovic,M.S.,** *On the generalization of some algorithms for the simultaneous approximation of polynomial roots,* in Interval Mathematics 1980 (K. Nickel Ed.), Academic Press, New York, 1980.

[Pet---81] **Petkovic,M.S.,** *On a generalization of the root iterations for polynomial complex zeros in circular interval arithmetic,* Computing **27**(1981)37–55.

[Pet---82] **Petkovic,M.S.,** *On an iterative method for simultaneous inclusion of polynomial complex zeros,* J. Computational and Appl. Math. **8**(1982)51–56.

[PetM--83] **Petkovic,M.S., Milovanovic,G.V.,** *A note on some improvements of the simultaneous methods for determination of polynomial zeros,* J. Computational and Appl. Math. **9**(1983)65–69.

[PetS--85] **Petkovic,M.S., Stefanovic,L.V.,** *On the simultaneous method of the second order for finding polynomial complex zeros in circular arithmetic,* Freiburger Intervallberichte **85/3**(1985).

[PetS--86a] **Petkovic,M.S., Stefanovic,L.V.,** *On a second order method for the simultaneous inclusion of polynomial complex zeros in rectangle arithmetic,* Computing **36**(1986)249–261.

[PetS--86b] **Petković,M.S., Stefanovic,L.V.,** *On some improvements of square root iteration for polynomial complex zeros,* J. Comp. and Appl. Math. **15**(1986)13–25.

[PetS--87] **Petkovic,M.S., Stefanovic,L.V.,** *On some iteation functions for the simultaneous computation of multiple complex polynomial zeros,* BIT **27**(1987) 111–122.

[Ral---81] **Rall,L.B.,** *Automatic differentiation: techniques and application,* Springer-Verlag, Berlin, 1981.

[Rok---73] **Rokne,J.,** *Automatic errorbounds for simple zeross of analytic functions,* Comm. ACM **16**(1973)101–104.

[RokL--71] **Rokne,J., Lancaster,P.,** *Complex interval arithmetic,* Comm. ACM **14** (1971)111–112.

[RokL--75] **Rokne,J., Lancaster,P.,** *Algorithm 86, complex interval arithmetic,* The Computer Journal **18**(1975)83–85.

[She---85] **Shearer,J.M.,** *Interval methods for Nonlinear Systems,* Ph.D. Thesis, University of St. Andrews, 1985.

[SheM--49] **Sherman,J., Morrison,W.J.,** *Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix,* Annals of Mathematical Statistics, **20**(1949)621.

[SheW--85a] **Shearer,J.M., Wolfe,M.A.,** *ALGLIB, a simple symbol— manipulation package,* Comm. ACM **28**(1985)820–825.

[SheW--85b] **Shearer,J.M., Wolfe,M.A.,** *The ALGLIB package,* Department of Applied Mathematics, University of St. Andrews, Fife, KY16 9SS Scotland, 1985.

[SheW--85c] **Shearer,J.M., Wolfe,M.A.,** *Some computable existence, uniqueness, and convergence tests for nonlinear systems,* SIAM J. Numer. Anal. **22/6** (1985)1200–1207.

[SheW--85d] **Shearer,J.M., Wolfe,M.A.,** *An improved form of the Krawczyk-Moore algorithm,* Appl. Math. Comput. **17**(1985)229–239.

[SheW--86] **Shearer,J.M., Wolfe,M.A.,** *A note on the algorithm of Alefeld and Platzöder,* SIAM J. Sci. Statist. Comput. **7**(1986)362–369.

[SheW--87] **Shearer,J.M., Wolfe,M.A.,** *Symbol Manipulation Packages,* Marcel Dekker Encyclopedia of Computer Science and Technology, (to appear), 1987.

[Sis---82a] **Sisser,F.S.,** *A modified Newton's method for minimizing factorable functions,* Journal of Optimization Theory and Application **38**(1982)461–482.

[Sis---82b] **Sisser,F.S.,** *Computer-generated interval extensions of factorable functions and their derivatives,* Intern. J. Computer Math. **10**(1982)327–336.

[Sis---82c] **Sisser,F.S.,** *Inverting an interval Hessian of a factorable function,* Computing **29**(1982)63–72.

[WanW--87] **Wang,D., Wu,Y.,** *Some modifications of the parallel Helley iteration method and their convergence,* Computing **38**(1987)75–87.

[Wei---03] **Weierstrass,K.,** *Neuer Beweis des Satzes, dass jede Ganze Rationale Function einer Veränderlichen dargestellt werden kann als ein Product aus Linearen Functionen darselben Veränderlichen,* Ges. Werke Vol. **3**(1903)251–269.

[Wol---78] **Wolfe,M.A.,** *Numerical Methods for Unconstrained Optimization,* Van Nostrand Reinhold Company, New York-Cincinati-Toronto-London-Melbourne, 1978.

[Wol---86] **Wolfe,M.A.,** *Private communication,* Departmant of Applied Mathematics, University of St.Andrews, KY16 9SS Scotland, 1986.

[Wol---87] **Wolfe,M.A.,** *Differentiation Arithmetic,* (Unpublished Manuscript), Department of Applied Mathematics, University of St.Andrews, KY 16 9SS Scotland, 1987.

[Woo---50] **Woodbury,M.,** *Inverting Modified Matrices,* Princeton University, Princeton, New Jersey, Statistical Research Group, Memorandum No.**42**, 1950.

[Van---78] **Vandergraft,J.S.,** *Introduction to Numerical Computations,* Academic Press, New York, 1978.

APPENDIX B

A Proof of Conjecture 3.8.1

By 3.8.15 and 3.8.18,

$$A^{(k-1)}y^{(k)} = c_{\cdot k}$$

whence by Cramer's rule

$$y_k^{(k)} = \frac{\sum_{i=1}^{n} c_{ik}A_{ik}^{(k-1)}}{\sum_{i=1}^{n} a_{ik}^{(k-1)}A_{ik}^{(k-1)}},$$

where $A_{ik}^{(k-1)}$ is the cofactor of $a_{ik}^{(k-1)}$ in the expansion of the determinant of $A^{(k-1)}$ by column k. So by 3.8.20 and 3.8.13,

$$\begin{aligned}\gamma^{(k)} &= \frac{\sum_{i=1}^{n}(a_{ik}^{(k-1)} + c_{ik})A_{ik}^{(k-1)}}{\sum_{i=1}^{n} a_{ik}^{(k-1)}A_{ik}^{(k-1)}} \\ &= \frac{det(A^{(k)})}{det(A^{(k-1)})}.\end{aligned}$$

Now by 3.8.14,

$$det(A^{(k)}) = \Delta^{(k)} \prod_{i=k+1}^{n} a_{ii}.$$

so

$$\gamma^{(k)} = \frac{\Delta^{(k)}}{\Delta^{(k-1)}a_{kk}}.$$

This proves Conjecture 3.8.1. □

APPENDIX C

A Proof of Conjecture 3.9.1

By definition of the $H^{(k-1)}$,

$$H^{(k-1)^{-1}} = \begin{pmatrix} B_U^{(k-1)} & O \\ B_L^{(k-1)} & I \end{pmatrix} \qquad (k = 1, \ldots, n),$$

where

$$B_U^{(k-1)} = \begin{pmatrix} b_{1,1} & \ldots & b_{1,k-1} \\ \ldots & & \\ b_{k-1,1} & \ldots & b_{k-1,k-1} \end{pmatrix}$$

and

$$B_L^{(k-1)} = \begin{pmatrix} b_{k,1} & \ldots & b_{k,k-1} \\ \ldots & & \\ b_{n,1} & \ldots & b_{n,k-1} \end{pmatrix}.$$

So

$$H^{(k-1)} = \begin{pmatrix} C^{(k-1)} & O \\ D^{(k-1)} & I \end{pmatrix}$$

where

$$C^{(k-1)} B_U^{(k-1)} = I$$

and

$$D^{(k-1)} B_U^{(k-1)} + B_L^{(k-1)} = O. \qquad C.1$$

So by 3.9.9,

$$y_k = \sum_{j=1}^{k-1} d_{kj}^{(k-1)} b_{jk} + b_{kk} \qquad C.2$$

where

$$D^{(k-1)} = \begin{pmatrix} d_{1,1}^{(k-1)} & \ldots & d_{k,k-1}^{(k-1)} \\ \ldots & & \\ d_{n,1}^{(k-1)} & \ldots & d_{n,k-1}^{(k-1)} \end{pmatrix}.$$

By $C.1$,

$$B_U^{(k-1)T} d_{k\cdot}^{(k-1)} = -b_{k\cdot U}$$

where

$$b_{k\cdot U} = (b_{k,1}, \ldots, b_{k,k-1}).$$

So by Cramer's rule,

$$d_{kj}^{(k-1)} = \frac{-det(B_{Uj}^{(k-1)T})}{\Delta_{k-1}}$$

where $B_{Uj}^{(k-1)T}$ is $B_U^{(k-1)}$ with row k replacing row j, and

$$\Delta_{k-1} = det(B_U^{(k-1)}).$$

So by *C*.2,

$$y_k = -\frac{1}{\Delta_{k-1}} \sum_{j=1}^{k-1} b_{jk} det(B_{Uj}^{(k-1)} + b_{kk}. \qquad C.3$$

Expanding down column k of $B_U^{(k)}$ we have

$$\begin{aligned} \Delta_k &= det(B_U^{(k)}) \\ &= -\sum_{j=1}^{k-1} det(B_{Uj}^{(k-1)}) b_{jk} + b_{kk}\, \Delta_{k-1}\,. \end{aligned}$$

So by *C*.3,

$$y_k = \frac{\Delta_k}{\Delta_{k-1}}.$$

This proves Conjecture 3.9.1. □